VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VYHLEDÁVÁNÍ PODOBNÝCH 3D MODELŮ

SEARCHING FOR SIMILAR 3D MODELS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **ZDENĚK ŠŤÁVA**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. MICHAL ŠPANĚL, Ph.D.**
SUPERVISOR

BRNO 2018

Zadání bakalářské práce/20891/2017/xstava08

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2017/2018

Zadání bakalářské práce

Řešitel: **Šťáva Zdeněk**
Obor: Informační technologie
Téma: **Vyhledávání podobných 3D modelů**
Searching for Similar 3D Models
Kategorie: Počítačová grafika

Pokyny:

1. Prostudujte dostupné materiály na téma popis polygonálních modelů a tvaru 3D objektů (tzv. shape descriptors). Seznamte se s metodami jejich sesouhlasení a porovnání.
2. Vyberte vhodné metody a navrhněte nástroj pro vyhledávání 3D modelů v databázi, kdy dotazem je ukázkový 3D model.
3. Experimentujte s vaší implementací a případně navrhněte vlastní modifikace metod.
4. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje.
5. Vytvořte stručný plakát nebo video prezentující vaši bakalářskou práci, její cíle a výsledky.

Literatura:

- Dle pokynů vedoucího.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Španěl Michal, Ing., Ph.D.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2017
Datum odevzdání: 16. května 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Tato práce se zabývá problematikou vyhledávání podobných 3D modelů v databázi obsahující až tisíce modelů. Věnuje se zejména porovnání již existujících deskriptorů užívaných pro popis 3D modelů a následné hodnocení podobnosti mezi modely. Porovnány jsou zejména deskriptory Rotation Invariant Spherical Harmonics a 3D Zernikeho deskriptor. Dále popisuje využívání knihoven na extrakci těchto deskriptorů a návrh různých experimentů s těmito knihovnami nad několika databázemi objektů. Zkoumá vliv škály, translace, deformace a rotace různých 3D modelů na výsledný deskriptor a celkovou přesnost obou vybraných metod. Tyto výsledky poté porovnává.

Abstract

This paper deals with searching similar 3D models in a database containing up to thousands of models. It focuses in particular on the comparison of existing descriptors used to describe 3D models and the subsequent evaluation of similarity between models. In particular, the descriptors Rotation Invariant Spherical Harmonics and 3D Zernike Descriptor are compared. It also describes the use of libraries to extract these descriptors and to design of various experiments with these libraries over several object databases. It examines the effect of scale, translation, deformation and rotation of different 3D models on the resulting descriptor and the overall accuracy of both selected methods. These results compare.

Klíčová slova

Vyhledávací engine, porovnávání, podobnost, 3D model, 3D tvar, 3D grafika, Spherical Harmonics, Zernike Descriptor, extrahování deskriptoru, experimenty

Keywords

Search engine, matching, similarity, 3D model, 3D mesh, 3D graphic, Shrerical Harmonics, Zerinke Descriptor, extracting descriptor, experiments

Citace

ŠŤÁVA, Zdeněk. *Vyhledávání podobných 3D modelů.* Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Michal Španěl, Ph.D.

Vyhledávání podobných 3D modelů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Michala Španěla, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Zdeněk Šťáva
16. května 2018

Poděkování

V této sekci je možno uvést poděkování vedoucímu práce Ing. Michalu Španělovi, Ph.D odbornou konzultaci a doporučení základních materiálů.

Obsah

Kapitola 1

Úvod

S nástupem moderní doby se lidé stále více a více snaží propojit reálný svět s virtuálním. Ať už to jsou brýle na virtuální realitu, či 3D obraz v televizních obrazovkách, nebo 3D počítačové hry. S nástupem nových technologií vznikají i nové požadavky, např vyhledávání modelů.

V dnešní době existuje mnoho vyhledávacích systémů založených na bázi textu, jako je např.Google. Některé moderní systémy dokáží vyhledávat ve 2D prostoru ve fotografiích.

Vyhledávání ve 3D prostoru není ale zatím stále dostatečně zmapováno, ač existuje spousta metod na toto vyhledávání. Ty se ale liší hlavně svojí přesností.

Tato práce se zabývá implementací vyhledávání na základě dvou metod a experimenty s nimi. Jde o metody *Rotation Invariant Sherical Harmonics* a *3D Zernike Descriptors*. Navrhovaný systém vyhledává jeden konkrétní model v databázi a zobrazí pro něj N nejpodobnějších modelů. Klíčové je, nad jakou databází se model vyhledává a zda jsou vyhledány očekávané modely. Poté je zhodnocena přesnost vybraných metod a výsledky se porovnají. Výsledný program tedy není určen ke koncovému uživateli, ale má pouze demonstrovat použití knihoven vytvořených v programovacím jazyce C++.

V kapitole č. 2 je popsaná teorie nutná k pochopení problematiky. Jsou zde nastíněny některé metody pro extrakci deskriptorů modelů, metody pro porovnávání modelů a metriky pro hodnocení kvality vyhledávání.

Kapitola č. 3 se zabývá stanovením cílů, samotným návrhem systému, jeho databází a návrhem testování.

V kapitole č. 4 je poté popsaná implementační část, tedy použité knihovny a nástroje. Dále je popsán výsledný script psaný v Pythonu.

V kapitole č. 6 jsou podrobněji popsané použité datové sady a zhodnocení dosažené kvality vyhledávání a návrhy na budoucí vylepšení.

Poslední kapitola č. 7 shrnuje celý výsledek práce.

Kapitola 2

Metody vyhledávání podobných modelů

Tato kapitola se věnuje teorii, která je dále využívána při návrhu nástroje a testování vybraných metod. Obsahem je obecné rozdělení deskriptorů a nastínění metod porovnávání těchto deskriptorů. Dále následuje podrobnější popis vybraných metod *Rotation Invariant Spherical Harmonics* a *3D Zernike descriptors*. Nakonec jsou popsány metody pro ohodnocení kvality vyhledávání.

2.1 Deskriptory modelů

Deskriptor modelů lze chápat jako popisovač tvaru modelu. Model je ve své základní formě nejčasněji reprezentován sítí polygonů. Teorie reprezentace modelů je blíže popsána v knize [6, str. 2–9].

Porovnávat modely v této formě by bylo velmi obtížné. Proto se ve vyhledávání volí přístup, kdy se pro každý porovnávaný model vypočítá deskriptor jeho tvaru a v této formě se následně deskriptory porovnávají. Deskriptory mohou mít různou formu reprezentace, avšak základním požadavkem pro každý z nich je, aby se lišily pro podobné modely méně než pro různé modely. Mezi další požadavky patří například invariantnost vůči rotacím modelu, či odolnost vůči posuvům a změně velikosti. Ne však všechny deskriptory tyto požadavky splňují.

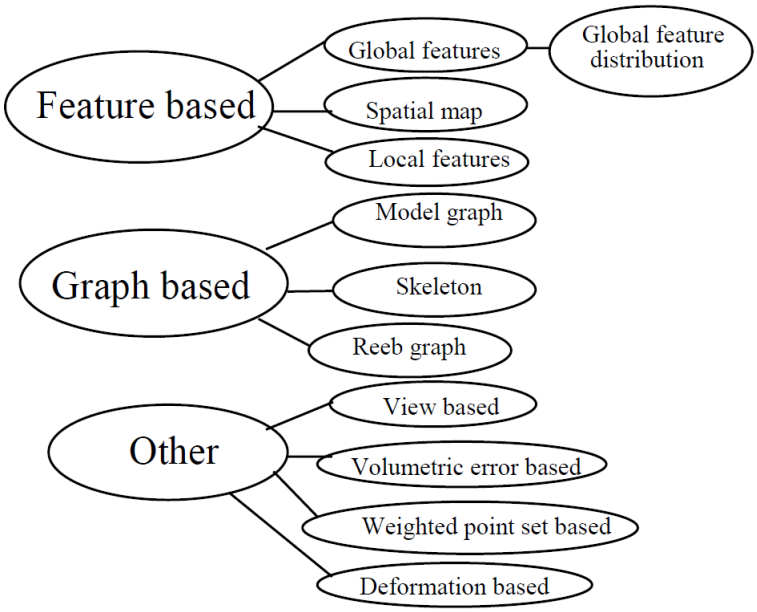
Podle formy reprezentace a jejich vlastností se dají deskriptory rozdělit do následujících kategorií:

- metody založené na vlastnostech
- metody založené na grafech
- metody založené na geometrii

2.1.1 Metody založené na vlastnostech

Tyto metody lze dále dělit podle typu vlastností, které popisují:

1. globální vlastnosti (*Global features*)
2. lokální vlastnosti (*Local features*)



Obrázek 2.1: Obrázek znázorňující rozdělení deskriptorů do kategorií [5].

3. metody založené na distribuci vlastností (*Global feature distribution*)
4. prostorová mapa (*Spatial map*)

Metody 1,2 a 4 mají pro model pouze jeden deskriptor, který je reprezentován n-dimenzionálním vektorem hodnot, kde n je pevná velikost vektoru pro všechny modely. Metody založené na lokálních vlastnostech mají určenou řadu povrchových bodů a deskriptor je počítán pro každý bod. Nejběžnější je reprezentace tvaru modelu pomocí jednoho globálního deskriptoru, který charakterizuje jejich celkový tvar. Avšak nedokáží zachytit konkrétní detaily tvaru.

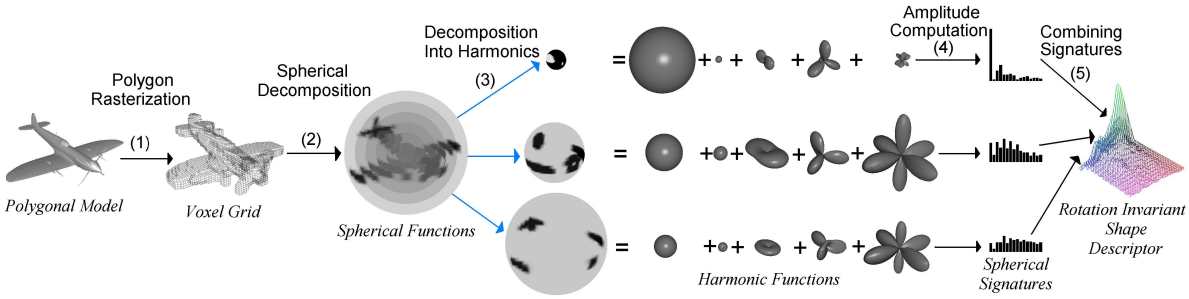
Metody založené na distribuci vlastností tuto myšlenku vylepšují. Prostorové mapy spíše zachycují prostorové umístění objektu. Jednotlivé mapy odpovídají fyzickým umístěním nebo úsekům objektu a jsou uspořádány způsobem, který zachovává jejich relativní polohy. Tyto metody ale nejsou invariantní k rotaci na rozdíl od metod 1,2 a 3. Výjimku tvoří speciálně navržené metody.

Spherical Harmonics

Metoda *Spherical Harmonics* je jedna z nejrozšířenějších metod pro extrakci deskriptorů. Řadí se do kategorie Spatial map. Neměla by být tedy invariantní vůči rotacím. Pro tuto metodu ale existuje několik rozšíření, které problém invariantnosti řeší. Např. *PCA-Spherical Harmonics Transform* (PCA-SHT) nebo *Rotation Invariant Spherical Harmonics* (RISH).

Autor ve svém díle [2] popisuje návrh RISH a jeho implementaci. Obrázek 2.2 zobrazuje postup extrakce deskriptoru, jenž využívá tato metoda. Extrakce deskriptoru se dá dekomponovat do 5 kroků:

1. Nejprve se rasterizuje polygonální povrch modelu do $2R \times 2R \times 2R$ voxelové mřížky.



Obrázek 2.2: Obrázek znázorňuje průběh metody spherical harmonics [1].

2. Voxelová mřížka se považuje za skutečnou funkci definovanou na množině bodů s délkou menší nebo rovnou R. Tato funkce je definovaná:

$$f(r,\theta,\phi) = Voxel(r\sin(\theta)\cos(\phi)+R, r\cos(\theta)+R, r\sin(\theta)\sin(\phi)+R) \tag{2.1}$$

kde $r \in [0;R], \theta \in [0;\pi] \wedge \phi \in [0;2\pi]$ a r značí poloměr.

3. Pomocí *spherical harmonics function* f_r vyjadřujeme každou funkci od součtu různých frekvencí:

$$f_l^m(\theta,\phi) = \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_l^m(\cos(\theta))e^{im\phi} \tag{2.2}$$

kde $m = -l, -(l-1), ..0, ..l-1, l$ a P_l^m jsou asociované Legendreho polynomy.

4. Výpočet normy $\|f_r^m\|$ každé složky.

5. Všechny normy se uloží ve formě vektorů.

3D Zernike descriptors

Metoda 3D Zernike descriptors [4] je dalším rozšířením metody Spherical Harmonics. Metoda je založena na *3D Zernikeho momentech*. Tyto momenty jsou vypočteny jako projekce dvou funkcí definující objekt na množině ortogonálních funkcí — 3D Zernikeho polynomech na jednotkovém kruhu.

2.1.2 Metody založené na grafech

Metody založené na grafu se podstatně liší od jiných deskriptorů založených na vektoru svou strukturou. Objekt postupně dekomponují na menší části. Díky tomu, že graf může reprezentovat vztahy mezi dvěma libovolnými částmi 3D objektu, jsou přesnější než výše popsané metody.

Tyto deskriptory jsou mnohem složitější a je obtížnější je vytvářet. Jejich realizace vyžaduje specializované systémy. Rozdělení metod a konkrétní metody jsou popsány v [6].

2.1.3 Metody založené na geometrii

I tyto metody lze dělit na další podsekce. Metody založené na pohledu mají jednoduchou myšlenku. Pokud je porovnávaný objekt stejný ze všech úhlů pohledu s dalším porovnávaným objektem, jsou oba objekty stejné. Avšak tyto metody jsou velmi paměťově náročné.

Dále existují metody využívající objemovou odchylku, sadu vážených bodů a metody založené na deformaci.

2.2 Metody porovnání deskriptorů

Porovnávání se liší v závislosti na tom, jaký druh deskriptoru zvolíme. Tato práce se v návrhu zabývá metodami založenými na vlastnostech popsaných v sekci 2.1.1, proto se následující odstavce budou věnovat problematice porovnávání deskriptorů reprezentovaných n-dimenzionálním vektorem.

Podobnost dvou modelů je definována jako vzdálenosti mezi vektory jejich deskriptorů [5]. Pokud jejich rozdíly odpovídají malé vzdálenosti, znamená to velkou podobnost. Formálně můžeme vzdálenost d na množině M definovat jako funkci:

$$d : M \times M \to R^{+} \cup \{0\} \tag{2.3}$$

kde d je vzdálenost. U funkce se požaduje splnění následujících vlastností:

1. **identita:** $\forall x \in M, d(x, x) = 0$
2. **pozitivita:** $\forall x \neq y \in M, d(x, y) > 0$
3. **symetrie:** $\forall x, y \in M, d(x, y) = d(y, x)$
4. **trojúhelníkovou nerovnost:** $\forall x, y, z \in M, d(x, z) \leq d(x, y) + d(y, z)$
5. **invariantnost vůči transformacím** Pro vybranou množinu transformací G platí, $\forall x, y \in M, g \in G, d(g(x), g(y)) = d(x, y)$

Podobnost p je poté nepřímo úměrná vzdálenosti d. Vypočítáme ji podle vzorce:

$$s = \frac{1}{1 + d} \tag{2.4}$$

Podobnost musí vždy nabývat hodnot z intervalu (0;1}.

2.2.1 Vzdálenost v Euklidovském prostoru

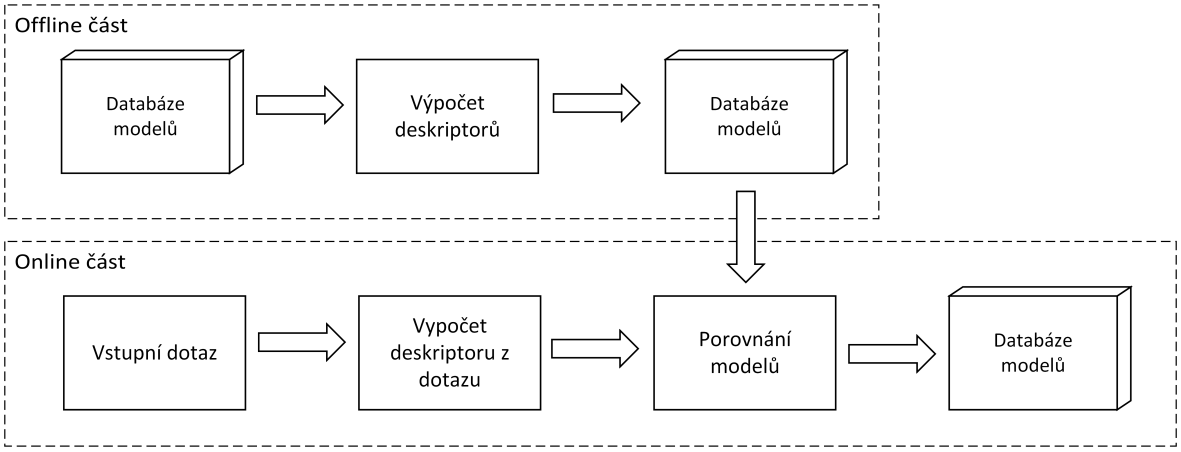
Euklidovský prostor je definován jako množina E^n prvků z R^n s metrikou d_E definovanou pro $X = (x_1, x_2, ...x_{n)} \in R^n$ a $Y = (y_1, y_2, ...y_n) \in R^n$ vztahem:

$$d_E(X, Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y + 2)^2 + ... + (x_n - y_n)^2} \tag{2.5}$$

Pokud funkce d_E je euklidovká metrika, pak číslo $d_E(X, Y)$ se nazývá **euklidovská vzdálenost**. Tato metrika splňuje první, druhou a třetí podmínku definovanou výše, je tedy vhodná pro použití vypočítání vzdálenosti dvou deskriptorů.

2.2.2 Kosinová podobnost

Kosinová podobnost se počítá na základě úhlu dvou nenulových vektorů, tzn. z obou vektorů se vypočítá kosinus jejich úhlu. Rozsah této metody se pohybuje v intervalu {-1;1}. Hodnota 1 a -1 znamená totožný vektor a hodnota 0 ortogonální, tedy opačný. Avšak případ porovnávání dvou modelů je specifický případ, ve kterém se výsledné porovnání pohybuje v intervalu {0;1}.



Obrázek 2.3: Znázornění principu vyhledávání.

Kosinus dvou vektorů vypočítáme podle vzorce:

$$s_k = cos(\theta) = \frac{A.B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B + i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} A_i^2}} \tag{2.6}$$

2.3 Vyhledávací metody

Existuje-li množina 3D modelů M a jeden model jako dotaz q, podobné modely z množiny se získají na základě podobnosti popsané v sekci 2.2. Pro tento účel je třeba znát podobnost dotazu se všemi prvky množiny M. Nejpodobnějších n modelů se poté získá seřazením všech podobností.

Jak je popsáno v sekci 2.1, porovnávání modelů v jejich základním tvaru není snadné. Proto se volí přístup, kdy se každý model zjednodušuje pomocí nějakého deskriptoru. Výhodou tohoto přístupu je fakt, že pro každý model stačí vypočítat deskriptor pouze jednou. Bylo by tedy velice neefektivní pro každý dotaz zjednodušovat vždy celou množinu modelů. Klasický přístup tedy vypočítá deskriptory z celé množiny modelů M na pozadí celého procesu a poté každý dotaz používá již vypočítaných deskriptorů.

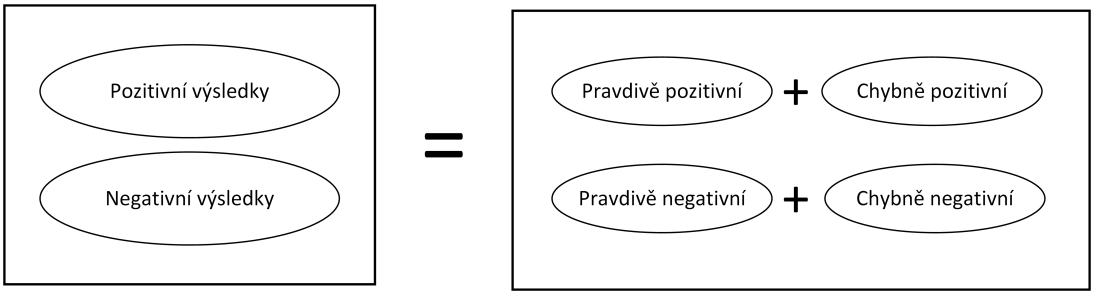
Na základě této teorie se operace vyhledávání rozdělují na dvě části:

- offline operace
- online operace

Jak již název napovídá, **offline operace** je ta část, která se děje na pozadí celého vyhledávání. Typicky se stará o správu databáze, její rozšiřování a o výpočet deskriptorů modelů. Tato část je nezávislá na dotazování a většinou se provádí jen jednou.

Online operace se provádějí při každém dotazu. Typicky zpracují vstupní model a porovnají jej s již vypočítanými deskriptory. Všechna porovnání poté seřadí a zobrazí n nejpodobnějších výsledků.

Celý problém znázorňuje obrázek 2.3



Obrázek 2.4: Obrázek znázorňující problém s množinami vyhledávání.

2.4 Metriky kvality vyhledávání

Metriky jsou důležitou částí pro vyhodnocení kvality vyhledávání. Vzhledem k různému vnímaní podobnosti různých lidí není možné stanovit triviální metriku.

Měřící funkce se rozděluji na dvě základní skupiny:

- pro binární hodnocení
- pro stupňované hodnocení

V našem případě je vyhledávání definováno jako binární klasifikace, tato práce se tedy dále zabývá jen metrikami pro binární hodnocení. Ostatní techniky rozvání autor ve své práci [3].

2.4.1 Rozdělení množin výsledku

Binární vyhledávání rozděluje množinu všech modelů M na dvě podmnožiny - na pozitivní a negativní výsledky, přičemž platí, že $M = P \cup N$, kde P je množina výsledků vyhledaných enginem a N je množina negativních výsledků. Musíme ale však předpokládat, že náš vyhledávací systém není dokonalý a že v množině P mohou být i negativní výsledky a v negativních pozitivní. Na toto toto rozdělení se dá nahlížet jako na matici čtyř klasifikovaných výsledků a přináší čtyři výstupy:

- **pravdivě pozitivní** (True positives)
- **chybně pozitivní** (False positives)
- **pravdivě negativní** (True negative)
- **chybně negativní** (False negative)

Pravdivě pozitivní (PP) jsou výsledky, které jsou vyhledány systémem a zároveň patří do množiny všech pravdivých výsledků P. *Chybně pozitivní (CHP)* jsou ty výsledky, které nejsou vyhledány, ale patří do množiny P. *Pravdivě negativní (PN)* jsou výsledky, které systémem nejsou vyhledány a zároveň patří do množiny negativních výsledků N a *chybně negativní (CHN)* výsledky jsou vyhledány jako pozitivní, ale ve skutečnosti patří do množiny N. Celý problém názorně ukazuje obr. 2.4

Při rozhodování, do kterých výsledků vyhledaný objekt patří, musí být známy správné výsledky vyhledávání. tzv predikce. To se v praxi děje klasifikováním samotných objektů do

tříd. Dva objekty patří do jedné třídy, pokud pro ně platí, že jsou si podobné. Např. existuje-li třída aut, patří do ní všechna osobní auta různých velikostí, typů a značek. Klasifikací modelů do tříd se také někdy označuje za předpovídání výsledků.

Klasický přístup rozdělování do množin P a N se hodnotí na základě prahové hodnoty t a ohodnocovaní funkce f:

$$P = \{x \in M, f(x) > t\} \tag{2.7}$$

$$N = \{x \in M, f(x) \leq t\} \tag{2.8}$$

Rozdělení do klasifikační matice poté probíhá na základě rozdělení modelů do tříd.

2.4.2 Základní definice

Definujeme dva základní požadavky na vyhledávací nástroje:

- nalézt všechny relevantní dokumenty
- nalézt pouze relevantní dokumenty

A právě tato dvě kriteria vyjadřují funkce *precision* a *recall*. Ty se nejčastěji používají pro hodnocení kvality vyhledávání.

$$PRC \ = \ \frac{pravdive\ pozitivni}{(pravdive\ pozitivni) \ + \ (chybne\ pozitivni)} = \frac{pravdive\ pozitivni}{pozitivni\ (mnozina\ P)} \tag{2.9}$$

Funkce *Precision* (PRC) vypovídá o tom, jak velká část ze všech možných relevantních výsledků byla nalezena. Vyjadřuje přesnost, nebo-li čistotu počtu nalezených výsledků. Pokud jej budeme chtít definovat, využijeme poznatků o dělení množin. Je definován jako poměr mezi počtem správně nalezených položek a celkovým počtem pravdivých výsledků.

$$RCL \ = \ \frac{pravdive\ pozitivni}{(pravdive\ pozitivni) + (chybne\ negativni)} \tag{2.10}$$

Funkce *Recall* (RCL) je zlomek relevantních výsledků mezi vyhledanými modely - tedy těch pravdivě pozitivních i těch chybně negativních.

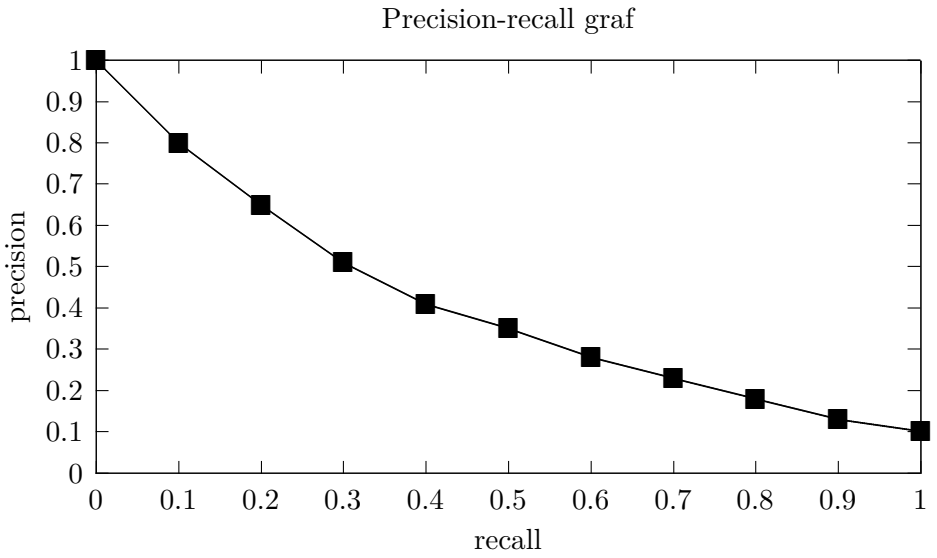
Další zajímavé hodnoty, které se dají na základě klasifikované matice měřit, jsou hodnoty *Error rate*, *Accuracy* a *false positive rate*.

Funkce *Error rate(ERR)*, nebo-li míra chyb, se vypočítá jako jako počet všech nesprávně vyhodnocených výsledků, děleno celkovým počtem datové sady. Určuje tedy chybovost měření:

$$ERR \ = \ \frac{(chybne\ pozitivni) \ + \ (chybne\ negativni)}{(pozitivni) \ + \ (negativni)} \tag{2.11}$$

Funkce *Accuracy (ACC)* měří přesnost vyhledávání trochu jiným způsobem než precision. Tato přesnost se vypočítá jako počet všech správně vyhledaných výsledků dělených celkovou velikostí databáze. Jde tedy o opačnou hodnotu k ERR.

$$ACC \ = \ \frac{(pravdive\ pozitivni) \ + \ (pravdive\ negativni)}{(pozitivni) \ + \ (negativni)} \ = \ 1 \ - \ ERR \tag{2.12}$$



Obrázek 2.5: Příklad precizion-recall grafu.

Funkce *False positive rate (FPR)* je míra počtu negativních výsledků v pozitivních výsledcích. Vypočíta se jako počet nesprávných pozitivních výsledků dělených celkovým počtem negativních výsledků.

$$FPR = \frac{chybne\ pozitivni}{(chybnepozitivni) + (pravdivenegativni)} \tag{2.13}$$

Opačná hodnota k FPR se nazývá *Specificity (SP)* a měří poměr správně určených negativních výsledků ku celkovému počtu negativních výsledků.

2.4.3 Precision-recall graf

Precision i recall spolu velmi úzce souvisí. Pokud bychom chtěli, aby precision byla rovna jedné, stačí, abychom vrátili pouze jeden výsledek u kterého předpokládáme, že je relevantní. Avšak tímto krokem by velmi klesla hodnota funkce recall. Kdybychom chtěli opačně maximalizovat hodnotu funkce recall, stačilo by vrátit všechny modely z databáze. I zde by opět hodnota precision klesla na minimum. Z tohoto důvodu se k hodnocení kvality vyhledávání používají obě funkce ve společném grafu.

Tento graf je možné použít jak k ohodnocení jednoho dotazu, tak k ohodnocení celého systému. Stačí naměřit výsledky pro precision v bodech, ve kterých je recall = {0,0;0,1;0,2; ... ;1,0} a pomocí interpolace vytvořit křivku. Výsledkem by měl být podobný graf jako na obrázku 2.5.

2.4.4 Mean average precision

Mean average precision (MAP) je dobrý prostředek, jak počítat precision přes K dotazů, kde výsledkem je pevně daný počet výsledků N. Pro každý dotaz se vypočítá jeho *average precision* (AP) tak, že se zprůměrují hodnoty precision vypočítané z pozice, na které byl načten:

$$AP = \frac{\sum_{i=1}^{N} PRC_i}{N} \tag{2.14}$$

MAP se poté vypočítá jako průměr těchto hodnot:

$$MAP = \frac{\sum_{i=1}^{K} AP_i}{K} \tag{2.15}$$

2.4.5 Matthews correlation coefficient

Matthews correlation coefficient (MCC) je další způsob jak hodnotit kvalitu vyhledávání. Toto měřítko je výhodné, pokud jsou třídy modelů rozdílných velikostí. I když se nejedná o často využívanou metodu, je přesto považována za jednu z nejlepších metrik. MCC lze vypočítat podle vzorce:

$$MCC = \frac{(pravdive\ pozitivni)(pravdive\ negativni) - (chybne\ pozitivni)(chybne\ negativni)}{\sqrt{(PP + CHP)(PP + CHN)(PN + CHP)(PN + CHN)}} \tag{2.16}$$

Výsledek může být prokázán jako správná mezní hodnota. MCC nabývá vždy hodnot v intervalu -1;1, kde 1 značí perfektní vyhledávání, 0 není lepší než náhodné vyhledávání a -1 označuje úplný nesouhlas vyhledávání.

Kapitola 3

Výběr metod a návrh experimentů pro jejich porovnání

Obsahem této kapitoly je shrnutí cílů této práce. Dále se zabývá návrhem požadavků na systém vyhledávání a na použité metody. Poslední dvě sekce se zabývají testovací sadou a průběhem testování. Tato kapitola využívá teorii popsanou v kapitole 2.

Tato práce se zaměřuje na experimentální použití již navržených metod. Cílem je prozkoumat jejich přesnost, invariantnost vůči rotacím, translacím, vliv deformace a následně naměřené výsledky porovnat.

3.1 Princip vyhledávání

Stanovit si základní představu o vyhledávacím systému je základ pro tuto práci. Primárním cílem této práce je myšlenka porovnání metod na vyhledávání modelů a experimenty s nimi. Sekundárním cílem je nastínit problém technicky znalému uživateli (ne koncovému uživateli) a poskytnout mu jednoduchý nástroj, kterým by si mohl experimenty vyzkoušet sám na svém počítači. Systém tedy musí umět následující akce:

- **Příprava databáze** - Nástroj si sám umí připravit libovolnou databázi modelů, jak je popsáno v kapitole 2.3. Neměl by nijak omezovat možnosti vstupní databáze možnosti by měly být omezeny pouze pracovním strojem.
- **Vyhledávání** - Vyhledávání podobných modelů je stěžejní funkcí tohoto systému. Existuje-li jeden libovolný vstupní model, nástroj musí být schopen z databáze vyhledat nejpodobnějších n modelů tím, že všechny mezi sebou porovná a seřadí jejich podobnosti.
- **Zobrazení výsledků** - Je samozřejmé, že uživatel bude chtít zobrazit výsledky v názorné podobě. Nejlepším způsobem, jak člověku předat informaci o modelu, je jeho zobrazení pomocí obrázků. Výsledný systém by tedy měl umět z nalezených n výsledků vygenerovat náhledy modelů.

Dále by systém měl splňovat tyto vlastnosti:

- **Flexibilita** - Při experimentálním použití je flexibilita důležitou vlastností. Nástroj by měl být snadný pro ovládání a měl by být připraven na časté změny v databázi, nebo dokonce změnu celé databáze.

Součástí flexibility je i možnost použití libovolného formátu vstupního modelu. Nástroj by měl být schopen si poradit se širokou škálou formátů modelů a sám je pomocí nějakého nástroje, či knihovny zkonvertovat a dále zpracovávat.

- **Automatické vyhodnocení** - Při této práci jsou důležité výsledky. Výsledný nástroj by tedy měl automaticky umět vypočítat metriky důležité pro srovnání použitých metod. Mezi důležité metriky patří precision-recall graf a MMC.

Jelikož se jedná o nástroj, který má pouze demonstrovat použití metod, nemusí se zaměřovat na optimalizaci, rychlost, či na způsob využívání databáze architekturou klient-server. Využívání tohoto přístupu by jen vedlo k problémům, které sebou nesou určitou režii a zapouzdření databáze před uživatelem.

I k ukládání deskriptorů není třeba využívat nějakého databázového systému. Ukládání každého deskriptoru zvlášť do textového souboru splňuje obě předchozí vlastnosti — flexibilitu a přehlednost. Uspořádáním textových souborů do složky umožňuje komukoliv přehledně nahlédnout do databáze. Dále může snadno smazat položky z databáze tím, že jednoduše smaže soubor, či přidávat položky přidáním nového souboru. Proto by měl být vektor deskriptoru v souboru reprezentován čitelně, např. čísla oddělená separátorem. Tento návrh odděluje čísla mezerou.

Díky vektorové reprezentaci a jednoduchému principu ukládání se do souboru s vektorem ukládá i jeho kontrolní součet délky a kódování, ve kterém je soubor uložen. Před porovnáním dvou deskriptorů se vždy ověří, zda je kontrolní součet stejný s délkou vektoru a zda jsou oba vektory stejně veliké, což je požadavek vybraných porovnávacích metod popsaných v sekci 2.2. V případě různé délky deskriptorů nemůže porovnávání dále pokračovat. Touto kontrolou se může předejít chybám, nebo neočekávanému chování.

3.2 Vybrané metody

Metod na popis modelu existuje celá řada (popsáno v [6]). Tento výběr se zaměří na tři základní požadavky:

- **Snadná dostupnost** - Vybrané metody musí být snadno dostupné pomocí již implementované knihovny. Tato knihovna musí být volně dostupná.
- **Časté použití** - Vybrané metody na extrakci deskriptorů by měly být často používané, aby jejich srovnání přineslo užitek jejich uživatelům.
- **Přesnost** - Vybraná metoda musí dávat dobré výsledky, neboť u metod se špatnými výsledky klesá i její četnost použití.

Některé deskriptory mohou být pomalé, ale velmi přesné, některé zase rychlé a méně přesné. Na základě výše definovaných požadavků se zhodnotí vlastnosti metod a získané zkušenosti s jejich použitím.

Metoda Spherical Harmonics patří mezi nejpoužívanější metody vůbec. Její nevýhoda je ale ta, že je neinvariantní vůči rotacím. Metody Rotation Invariant Spherical Harmonics [2] a 3D Zernike desktiptor [4] jsou rozšířením této metody. Jejich autoři popisují metodu jako více přesnou než jejich původní metoda a zároveň doplněnou o vlastnost, jakou je invariantnost rotace. Jelikož se jedná o dvě rozšíření stejné metody, zaměří se tato práce na porovnání těchto dvou metod.

Obě tyto metody extrahují deskriptor modelu ve formě vektoru, proto musí být vhodně zvolené metriky porovnání. Metrika založená na Euklidovské vzdálenosti je nejčastěji používanou metrikou porovnání dvou vektorů. Kosinová podobnost porovnává dva vektory na základě jejich úhlu. Podle její definice v sekci 2.2.2 je na první pohled zřejmé, že si vyžaduje více výpočetních operací než metrika založená na základě Euklidovské vzdálenosti. Obě metriky používají k porovnání jiné techniky. Vhodnost použití těchto porovnávacích metod u vybraných deskriptorů se zaměřuje kapitola 5.

3.3 Testovací datové sady

Vhodná datová sada 3D modelů je klíčová k úspěšnému vyhodnocení. Avšak jedna sada nestačí. Malá sada neotestuje metodu stejně jako sada velká. Výhodné je mít v sadách různý počet tříd. Každá třída je specifická tím, že obsahuje modely, které mezi sebou nějakým způsobem souvisejí. Na klasifikaci tříd existují nástroje, které pracují na základě podobnosti modelů. Dokáží tedy vytvořit např. třídu vlaků. Pro experimenty budou ale potřeba i obecnější třídy, např. třída nábytku.

Svým způsobem jde o druh indexace modelů, kde index reprezentuje celá složka databáze. Např. bude-li na vstupu model židle, v neindexované databázi budeme moct teoreticky dostat jakýkoliv výsledek, tedy i ten, kdy mezi n modely nebude žádná židle. V databázi nábytku se pravděpodobnost a rychlost vyhledání ostatních židlí rapidně zvyšuje.

K otestování bude vhodné použít tyto druhy databází:

1. **Velká databáze** - Na této databázi se ověří vyhledání modelu mezi velkým počtem modelů. Počet modelů v této databázi by se měl pohybovat kolem 1000 a měl by mít kolem 40 tříd.

2. **Středně velká databáze různých modelů** - Ověření přesnosti vyhledávání mezi středně velkou sadou sadou různých modelů. Měla by mít kolem 300 modelů rozdělených asi do 15 tříd.

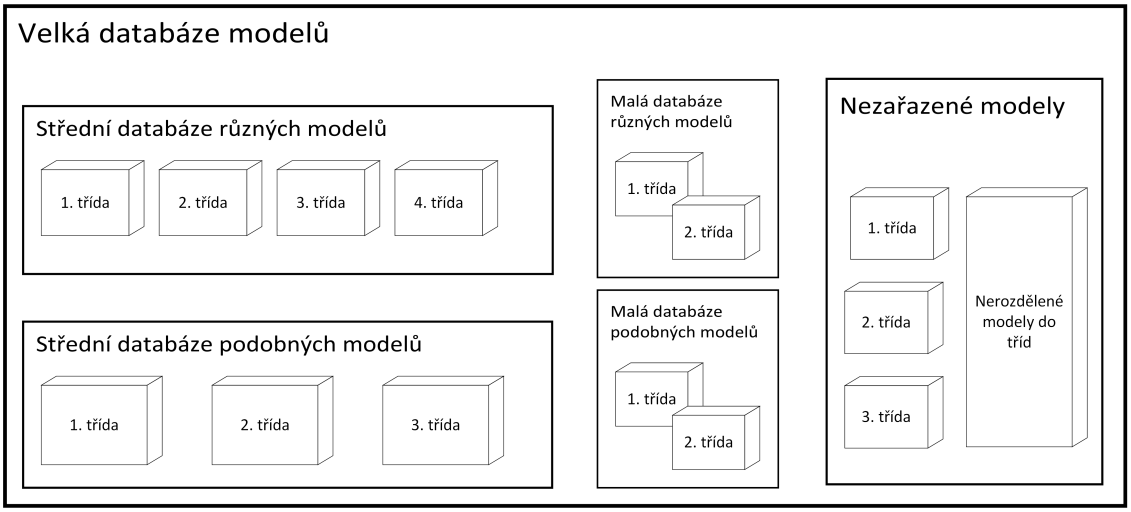
3. **Středně velká databáze podobných modelů** - Ověření přesnosti vyhledávání a chování metody mezi středně velkou sadou velmi podobných modelů, nebo modelů, které mezi sebou nějakým způsobem souvisejí. Počet modelů a tříd by měl být podobný jako v předchozím případě.

4. **Malá databáze různých modelů** - Ověření na malé sadě různých modelů. Měla by mít kolem 100 modelů rozdělených aspoň do 5 tříd.

5. **Malá databáze podobných modelů** - Ověření přesnosti na malé sadě velmi podobných modelů. Velikost databáze by měla být podobná jako v předchozím případě.

Pro účely testování bude tedy potřeba jedna velká datová sada složená z mnoha tříd a bude demonstrovat obecné využití metody, pokud bude uživatel náhodně vyhledávat v databázi na základě nějaké předlohy. Tato velká databáze se poté rozdělí na menší. U středně velkých databází bude rozdělení do tříd obecnější, než u malých databází.

Střední databáze různých modelů budou obsahovat střední počet tříd, které jsou svým způsobem nějak spojeny, ale nejsou si podobny. Vhodný příklad je databáze nábytku, která se dá rozdělit na třídu dveří, židlí, stolů, skříní, atd. Tento případ má demonstrovat příklad, kdy uživatel bude chtít vyhledávat nějaký kus nábytku na základě předlohy.



Obrázek 3.1: Znázornění rozdělení databáze.

Střední databáze podobných modelů obsahuje třídy, které jsou svým způsobem spojeny a zároveň jsou si velmi podobné. Typický příklad by mohla být databáze čtyřnohých zvířat, která by se dala rozdělit na psy, kočky prasata, atd. Tento druh databáze má demonstrovat použití metody v extrémních podmínkách, neboť prase a pes si mohou být vizuálně velmi podobné.

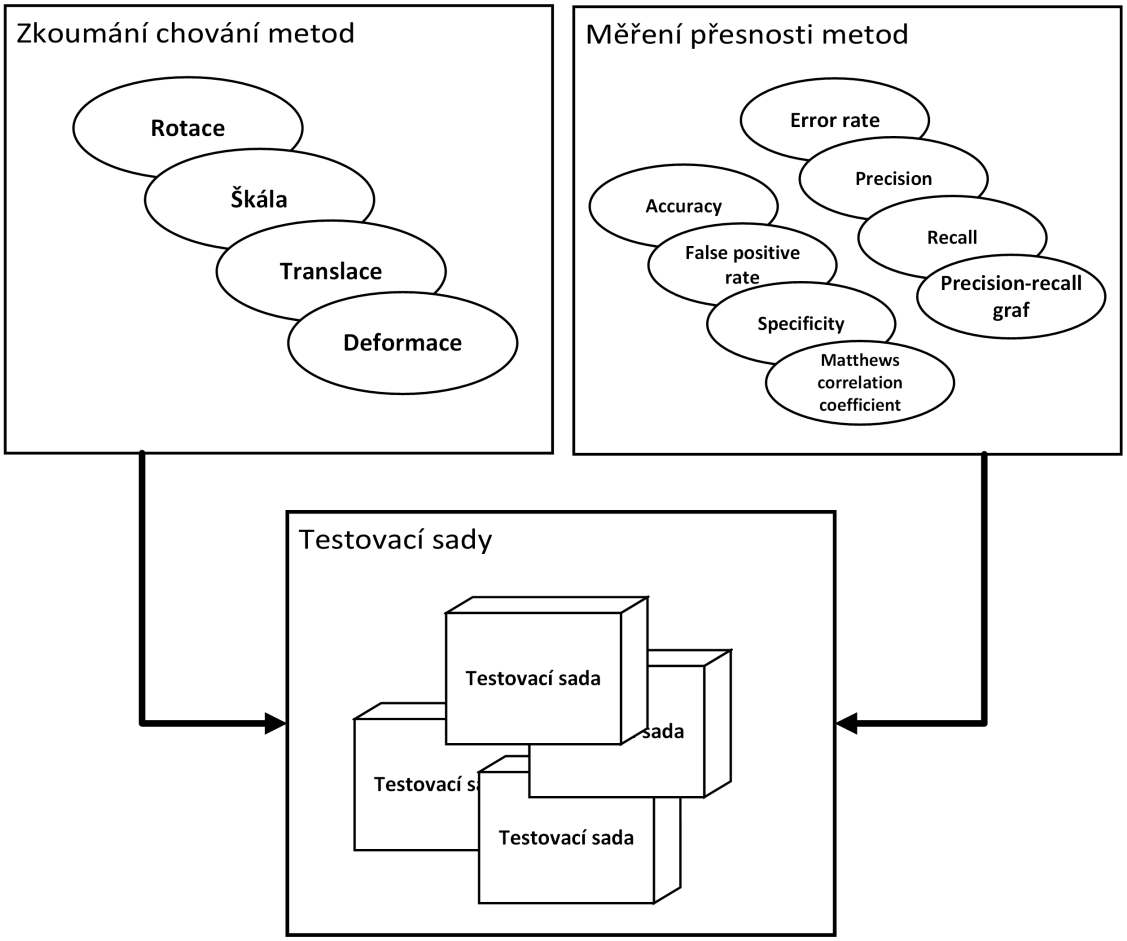
Stejným způsobem jsou navrhnuty malé databáze, s tím rozdílem, že se bude demonstrovat použití metod pro malý počet tříd. Bylo by ovšem vhodné, aby použité třídy nebyly obsaženy ve středně velkých databází.

3.4 Průběh testování

Testování na základě experimentů je hlavní náplní této práce. Snahou je co nejvíce otestovat obě vybrané metody v typických i extrémních případech a vyhodnotit je metrikami popsanými v kapitole 2.4. Testování bude probíhat na sadách splňující požadavky z předchozí sekce.

Autoři [2] a [4] vybraných metod ze sekce 3.2 slibují invariantnost vůči rotacím a translacím. Kromě těchto vlastností se bude testovat i chování metod v případě deformace modelu.

- **Invariantnost vůči rotaci** - Z každé datové sady se vezme jeden objekt, který se otočí kolem jeho os. Výchozí hodnota každého modelu je pro osu x, y a z je 0. Změnou těchto hodnost se vzhled modelu nezmění. Avšak metoda jej nemusí vyhodnotit stejně jako originální model.
- **Odolnost vůči translacím** - Původní model je vykreslený na osách x, y a z v bodech [0,0,0]. Tento test zjistí, zda se změní pohled vyhledávací metody na model, pokud se tento počáteční bod přesune na jiné místo v souřadném systému.
- **Odolnost vůči škále modelu** - U modelu z databáze se vezme objekt, u kterého se změní velikost škálováním po osách. Škálováním se model deformuje, avšak detaily zůstávají zachovány.
- **Deformace objektu** - Mírná deformace objektu jistě změní výsledky vyhledávání. To je logicky žádoucí efekt vyhledávání — pro každý jiný tvar, jiné výsledky. Tento



Obrázek 3.2: Znázornění průběhu testování vybraných deskriptorů.

test má za úkol prozkoumat, jak moc se změní pohled vyhledávací metody na model, pokud se jen nepatrně změní jeho vzhled. Deformací se myslí odebrání/přidání části modelu.

Všechny popsané testy se vyhodnotí porovnáním s původním modelem. V rámci těchto testů se otestuje i vliv porovnávacích metod, popsaných v sekci 2.2, na vyhledávání.

Kapitola 4

Implementace

Tato kapitola obsahuje popis implementace sady, která řeší výše popsaný návrh řešení.

Jelikož se nejedná o aplikaci, která směřuje ke koncovému uživateli, postačí si implementační řešení s jednoduchou konzolovou aplikací. Pro tento účel se jeví nejvhodněji použití dvou technologií, a to staticky kompilované C++ a dynamický Python, který slouží ke spojení a ovládání kompilovaných částí. Výsledkem tedy není jedna jediná aplikace, ale sada kompilovaných knihoven a tři Python skripty, z niž jeden spojuje všechny části dohromady. Oba vedlejší skripty jsou přes hlavní skript předávány nástroji Blender. První z nich se stará o generování náhledů ve formě obrázků a druhý o modifikaci modelů, důležitých pro experimentování s metodami. Celá sada byla vyvíjena na platformě Windows s 64-bitovou architekturou a byla použita verze Pythonu 3.6.3.

4.1 Použité knihovny

Při implementaci hrají použité knihovny klíčovou roli. Konkrétně jsou použity dvě knihovny pro extrakci deskriptorů, a to pro deskriptory *Rotation Invarian Spherical Harmonics*[1] a *3D Zernike deskriptors*[2]. Jak již bylo zmíněno, obě knihovny jsou implementovány v jazyce C++ a obě jsou veřejně dostupné.

První knihovna pro deskriptor Rotation Invariant Spherical Harmonics sama načítá model ve formátu Polygon File Format (ply) a výsledný deskriptor uloží do zvoleného souboru. Tato knihovna si nevyžadovala žádnou úpravu rozhraní, proto jsem použil již kompilovanou verzi od autora[3], protože byla mnohem rychlejší než mnou kompilovaná verze. Avšak autorem kompilovaná verze musí mít k dispozici externí knihovnu FFTW, kde byla použita nejnovější kompilovaná verze 3.3.5 pro 64-bitovou architekturu[4].

Druhá knihovna pro 3D Zernike descriptors pracuje s modely v binární 3D voxelové mřížce (binvox) a vyžadovala si úpravu rozhraní. Upravil jsem rozhraní tak, aby systém předávání vstupních informací a systém ukládání deskriptoru byl jednotný s první knihovnou. Tím jsem velmi zjednodušil práci s knihovnami a zvýšil čitelnost skriptu. Tato knihovna si nevyžaduje přítomnost žádných dalších knihoven ani nástrojů.

[1] github.com/mkazhdan/ShapeSPH
[2] github.com/Sunwinds/ShapeDescriptor
[3] htmlpreview.github.io/?https://github.com/mkazhdan/ShapeSPH/blob/master/descriptors.html
[4] www.fftw.org/install/windows.html

4.2 Využívané nástroje

Součástí sady jsou i dva nástroje, *meshconv*[5] a *binvox*[6], pro konverzi formátu modelů a nástroj *Blender*[7] pro generování náhledů modelů.

Nástroj *meshconv* se využívá pro konverzi modelů z databáze do formátu ply, který je dále využíván nejen ostatními nástroji, ale hlavně knihovnou pro extrakci deskriptoru Spherical Harmonics. Tento nástroj je velmi jednoduchý na používání. Umí převádět velkou řadu formátů (např. obj, ply, off, 3ds, ad.).

Nástroj *binvox* je nástroj, který čte soubor 3D modelu a rasterizuje ho do binární 3D mřížky. Výsledkem je stejnojmenný formát, který využívá knihovna pro extrakci Zernikeho deskriptoru. Díky tomu, že je od stejného autora, jako nástroj meshconv, je jeho použití velmi podobné a také velmi jednoduché.

Pro generování náhledů obrázků byl zvolen nástroj *Blender*, jež dovoluje i pokročilejší operace skrze předaný skript v jazyce Python. Nástroji se předá pole názvů N modelů a on postupně vygeneruje náhled pro každý z nich. Informace o kameře, světlu a kontextu prostoru jsou definovány přes skript. Ve skriptu je jednoduchý cyklus, který do scény přidává a odebírá modely, ze kterých je generovaný náhled.

Nástroj *Blender* se stará i o modifikaci modelů pro otestování vlastností deskriptorů. Nástroji se předá cesta ke složce s modely a název akce, kterou má provést. Jednoduchý cyklus ve skriptu projde všechny modely ve složce a udělá na nich požadovanou úpravu.

4.3 Popis činnosti hlavního scriptu

Jak již bylo řečeno, skript psaný v Pythonu je páteří celého procesu. Skript je zaměřen hlavně na přehlednost kódu, aby se v něm každý snadno vyznal. V hlavičce scriptu se nachází několik globálních proměnných, který určují chod celého skriptu. Celá myšlenka spočívá v tom, že pokud by někdo chtěl změnit chování programu, nemusí složitě měnit část kódu, ale stačí změnit jen hodnotu v globální proměnné. Např. se zde nachází proměnná, která určuje, kolik N výsledků se má zobrazit, či jaké formáty 3D modelů mají být zpracovávány.

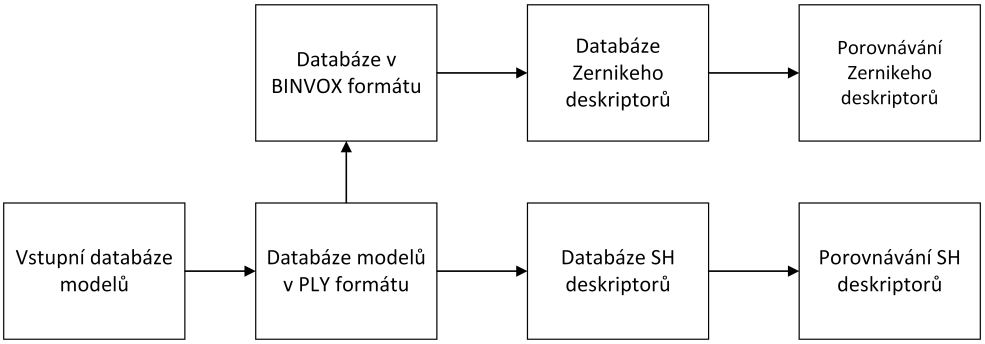
Celý skript je dále rozdělen na části, kdy každá část může fungovat odděleně, avšak části na sebe navazují. Např. bez dřívější konverze modelů z databáze na formát ply nelze vypočítat Rotation Invariant Spherical Harmonics deskriptor. Části se spouštějí pomocí argumentů příkazového řádku. Více informací o ovládání programu viz. Příloha A

4.3.1 Závislost částí skriptu a optimalizace

Hlavní skript byl navrhnut jako oddělené části z důvodu očekávané vysoké časové náročnosti procesu přípravy databáze modelů. Díky možnosti spouštění jednotlivých částí nezávisle na sobě, se mohou jednotlivé konverze mezi modely spouštět v různé časy a nemusí se pro velké databáze čekat na dokončení všech operací najednou. Skript samozřejmě podporuje i předpřipravení databáze najednou. Pro menší databáze není třeba spouštět složitě části po sobě. Obecně platí, že každá část pracuje pouze se dvěma formáty modelů. Většinou se jedná o konverzní funkce.

První část se zabývá prohledáním vstupní databáze a převedení všech akceptovatelných formátů do formátu ply. Za tímto účelem je vytvořena nová složka, do které se konvertované

[5] www.patrickmin.com/meshconv/
[6] www.patrickmin.com/binvox/
[7] www.blender.org



Obrázek 4.1: Obrázek popisující závislosti částí skriptu.

modely ukládají. Je to z toho důvodu, že vstupní databáze většinou bývá ve stromové struktuře a opakované prohledávání všech adresářů zbytečně zabírá mnoho výpočetního času.

Další část se zabývá převodem modelu do rasterizované mřížky. Tato část využívá již předpřipravený adresář s modely ve formátu ply. Výstup se poté ukládá do stejné složky. Další dvě části využívají již zkonvertovaných modelů a používají knihovny pro výpočet deskriptoru. Oba deskriptory se ukládají odděleně do složek z důvodu větší přehlednosti a faktu, že prohledávání adresáře i s modely by vedlo k značnému zpomalení porovnávání.

Poslední dvě části řeší samotné porovnávání vyhledávaný model převedou do požadovaného formátu a poté sestaví slovník, kde je název modelu z databáze a shoda se vstupním modelem. Vztahy mezi jednotlivými částmi jsou nastíněny v obr. 4.1.

Součástí skriptu je i funkce, která se stará o generování náhledů výsledků vyhledávání ve formě obrázku a funkce počítající metriky měření kvality vyhledávání.

Skript obsahuje základní konstrukce pro ošetření chybného vstupu, či nekorektního chování programu. Např. pokud by někdo chtěl porovnávat neexistující model, bude na tuto skutečnost upozorněn vhodnou chybovou hláškou a činnost skriptu bude ukončena. Stejné chování je nastaveno i na případy, kdy databáze neexistuje, model nelze číst, konverze neproběhla správně, či je chybný kontrolní součet velikosti deskriptoru nebo dva porovnávané deskriptory mají různou délku.

4.3.2 Získávání metrik

Součástí skriptu jsou funkce, které automatickou analýzou databáze dokáží získat metriky popsané v části 2.4. Tyto funkce počítají metriku buď pro prvních N výsledků, nebo pro prahovou hodnotu, nebo pro sadu těchto hodnot. Pro získání přesnějších výsledků, každá funkce iteruje přes celou databázi a pro každý model vypočítá požadovanou metriku. Výsledná hodnota se poté zprůměruje počtem modelů v databázi. Naměřené hodnoty se ukládají do souboru.

Kapitola 5

Popis výsledků experimentů

Tato kapitola popisuje výsledky experimentů na datových sadách. Zabývá se rozborem výsledků a porovnání metod. Experimenty navržené v kapitole 3.4 jsou postupně prováděny na datových sadách a pro každý experiment je zobrazen výsledek. K ohodnocení kvality se používají metriky popsané v kapitole 2.4.

5.1 Rozdělení tříd a použité modely

Základ pro datové sady byla použita knihovna *Princeton Shape Benschmark*[1], která obsahuje 1813 modelů. Tato knihovna byla rozdělena na několik menších databází ručním klasifikováním modelů. Třídy každé databáze se dají shrnout jednou nadtřídou, a splňují podmínky navržené v sekci 3.3.

Jako středně velká databáze, co se týká počtu tříd, byly zvoleny databáze *dopravních prostředků* a *nábytku*. Jako menší databáze jsou zvoleny třídy *rostlin* a *hlav*. Pro adekvátní porovnání naměřených metrik byly střední a malé třídy upraveny na stejnou velikost a na stejný počet tříd. Liší se tedy pouze podobností modelů.

Datová sada nábytku

Tato sada je pojmenovaná *Furnitude_11*. Zde byla snaha vytvořit třídu, která by obsahovala větší počet tříd, které si mezi sebou nebudou vizuálně podobny. Tato datová sada se skládá ze 130 modelů rozdělených do 11 tříd. Jsou zde obsaženy modely barových židlí, dveří, kulatých stolů, atd. Složení databáze názorně ukazuje obrázek 5.1.

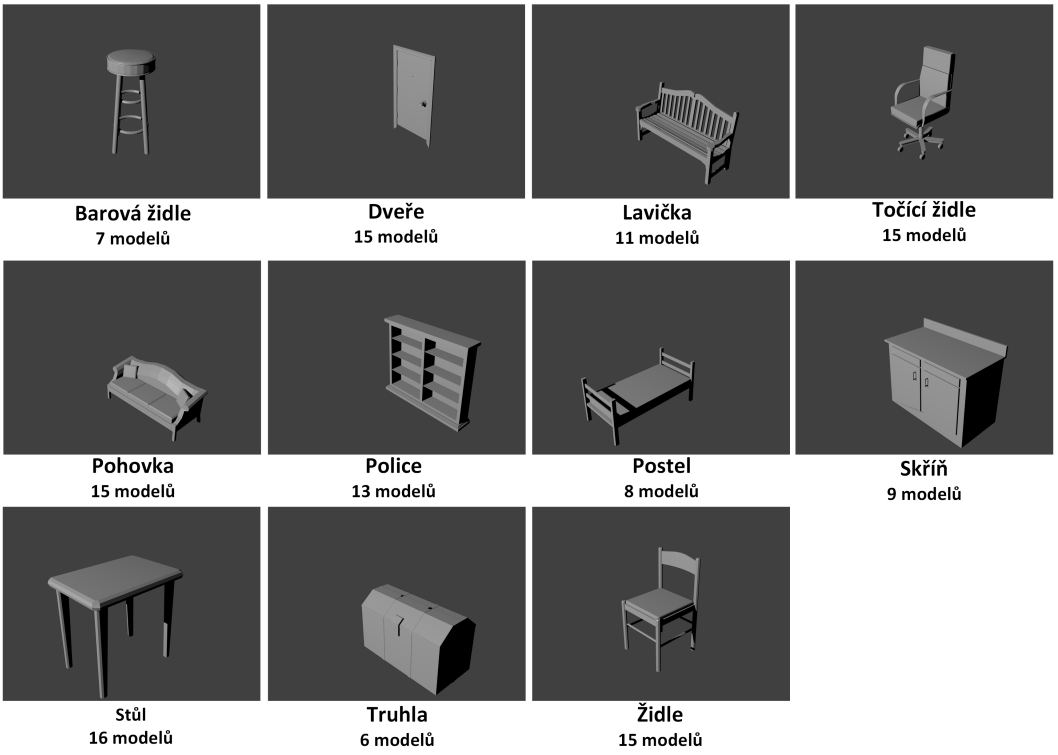
Tato datová sada má simulovat obecné použití metod na obecných databázích, které neobsahují mnoho podobných modelů.

Datová sada dopravních prostředků

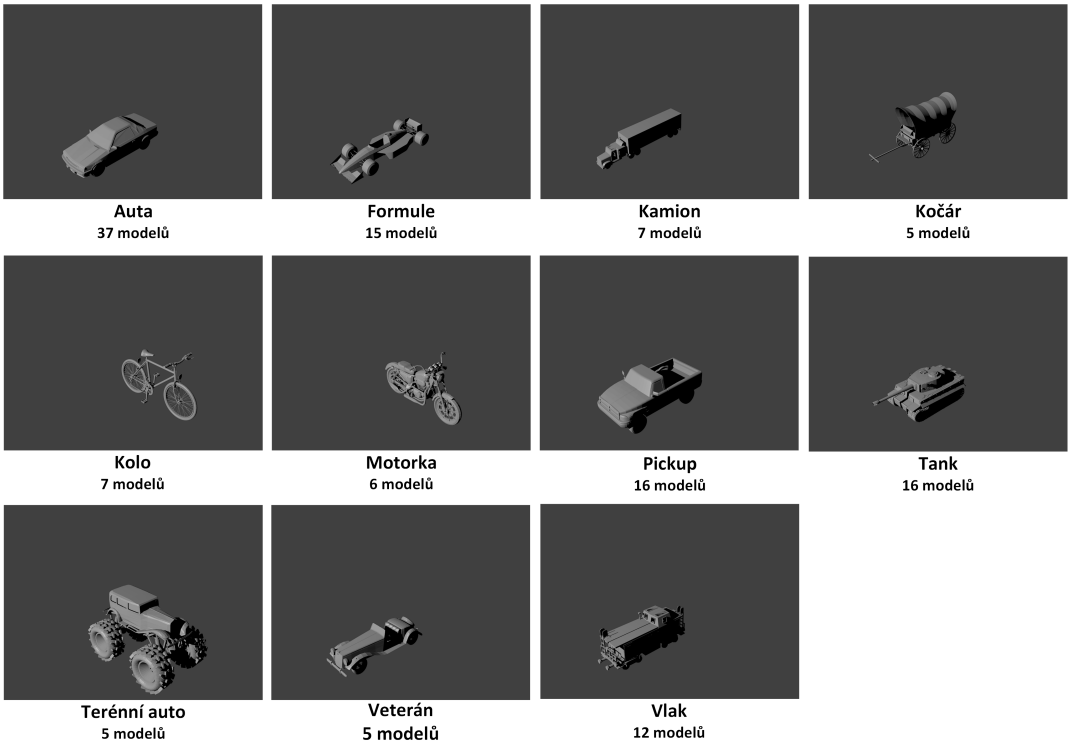
Sada byla pojmenovaná *Vehicles_11*. Svým složením je datová sada podobná předchozí datové sadě s tím rozdílem, že třídy v ní jsou si mnohem více podobné. Sada se skládá ze 130 modelů rozdělených do 11 tříd. Obsahuje třídy jako jsou například, vlak, tank, klasické auto, atd. Rozdělení databáze i s počtem modelů a tříd zobrazuje obrázek 5.2.

Tato datová sada má za úkol demonstrovat vyhledávání v extrémních podmínkách, kdy jsou si všechny objekty v databázi velmi podobné.

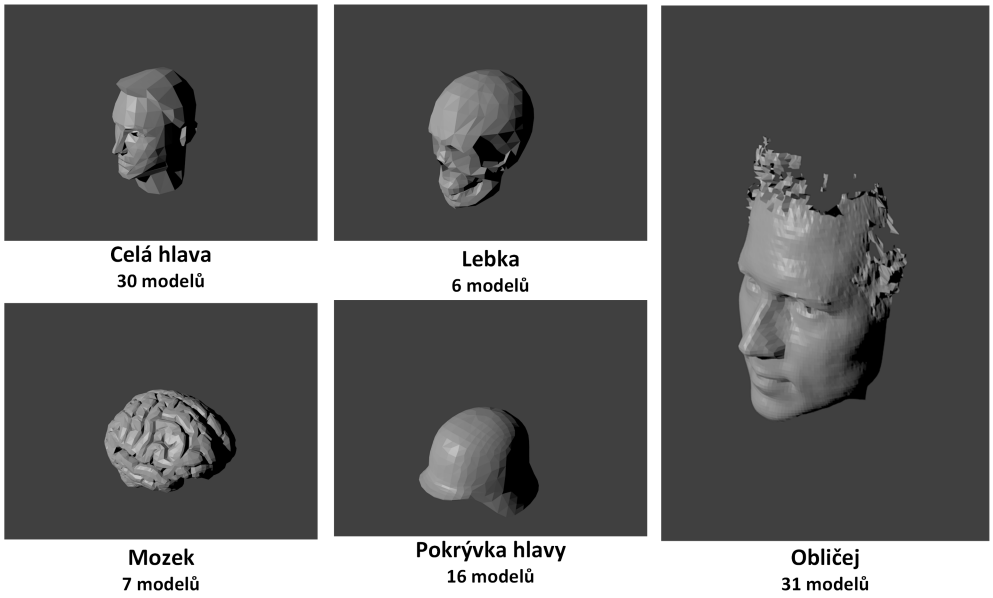
[1] 3d.csie.ntu.edu.tw/ dynamic



Obrázek 5.1: Složení databáze Furnitude_11.



Obrázek 5.2: Složení databáze Vehicles_11.



Obrázek 5.3: Složení databáze Heads_5.

Datová sada částí hlavy

Datová sada je podobná sadě nábytku, jen menší. Je pojmenovaná podle toho, že obsahuje části lidské hlavy a nese označení *Heads_5*. Skládá se z 88 modelů rozdělených do 5 tříd. Z obrázku 5.3 zle vyčíst složení databáze. Můžeme zde najít mozek, lebku, nebo třeba jen část obličeje. Všechny modely mají sice kulatý tvar, ale jejich detaily jsou jiné.

Datová sada rostlin

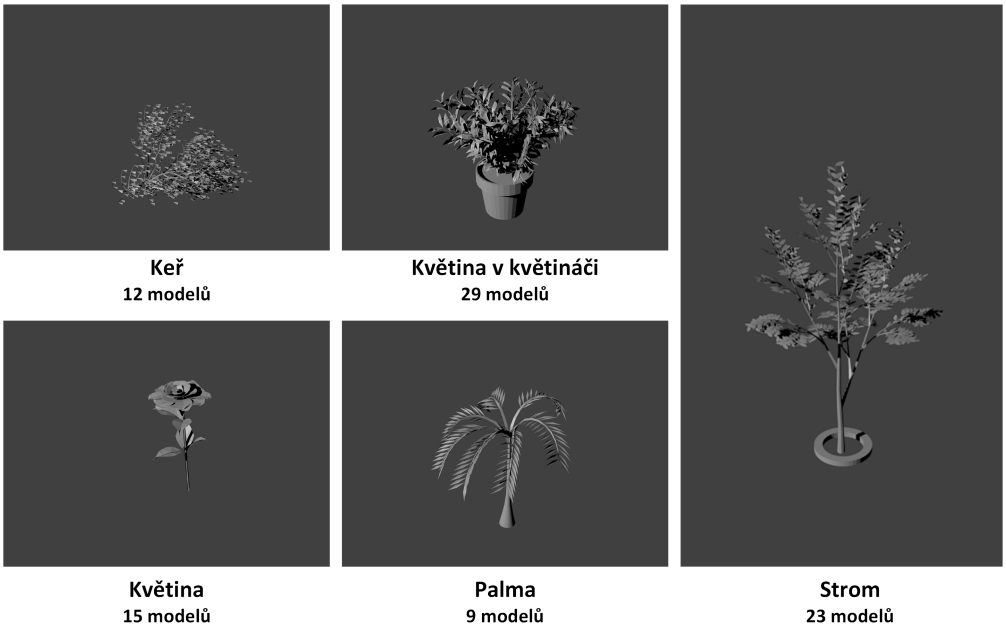
Tato datová sada byla pojmenovaná *Plants_5*. Všechny třídy jsou si velmi podobné, jde tedy o případ dalšího extrémního testování na 88 modelech rozdělených do 5 tříd. Celou situaci zobrazuje obrázek 5.4. Jsou zde např třídy květina, palma, nebo třída ostatních stromů bez palem.

Velká datová sada

Tato datová sada obsahuje všechny modely z datových sad popsaných výše a nějaké další. Mezi další modely patří modely nábytku, či rostlin. Demonstrovat má hlavně vliv tříd modelů na jednotlivé metody. Celkem tedy obsahuje 516 modelů rozdělených do 35 tříd a nese označení *Database_35*.

Modely na testování vlastností

Pro otestování invariantnosti rotace, škály a translace modelu je potřeba mít k dispozici upravené modely. Tyto testy se budou provádět z důvodu přesnějších výsledků na 100 náhodně vybraných modelech z původní databáze, které se následně několikrát upraví. Všechny úpravy jsou provedeny dvakrát, jednou představují menší modifikaci modelu, podruhé větší. Menší modifikace zde představuje změnu jen na jedné ose, větší změna je provedena na všech osách. Následně se bude testovat, jak jsou upravené modely podobny původnímu modelu.



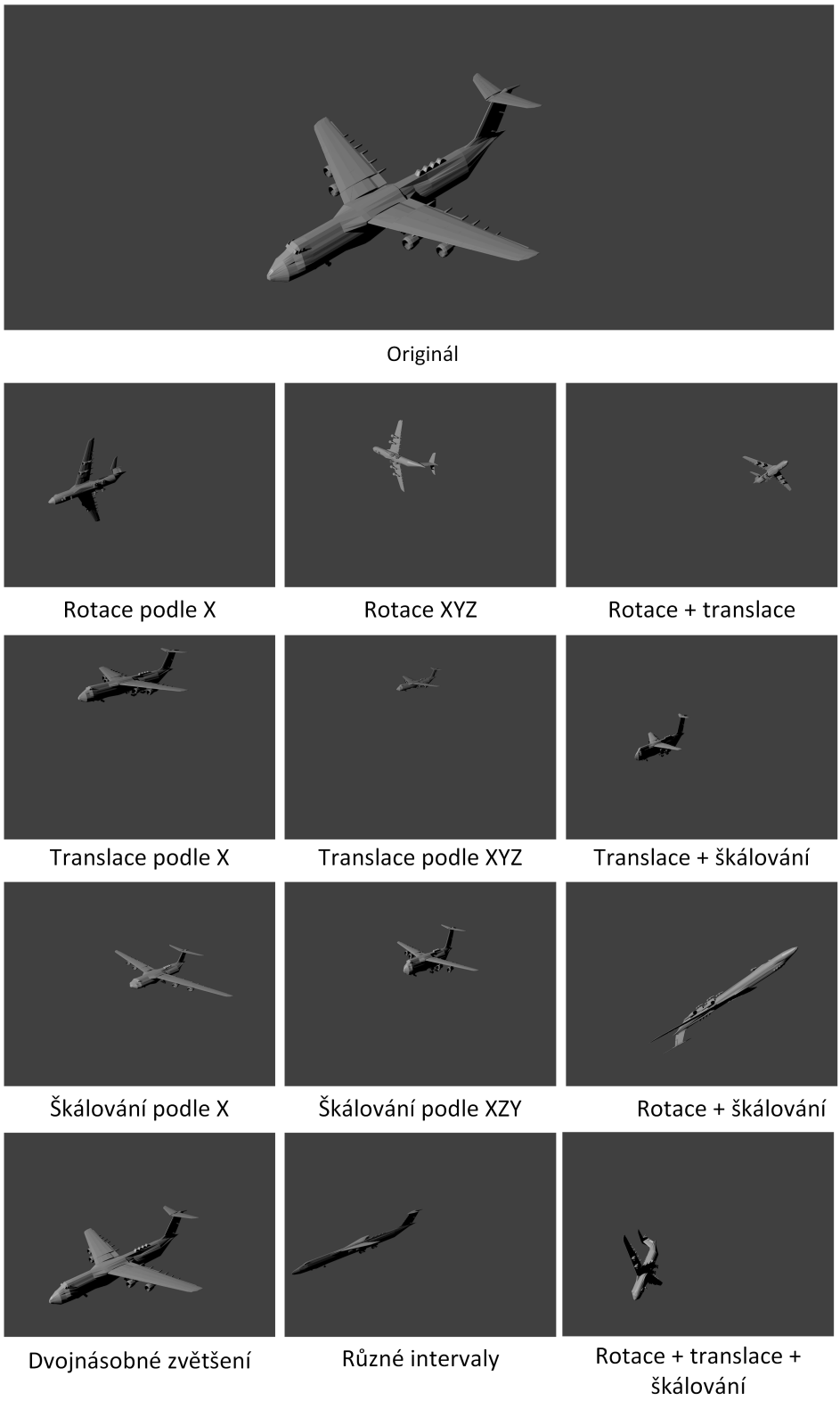
Obrázek 5.4: Složení databáze Plants_5.

K tomuto experimentu nejsou potřeba klasifikace, ani jiné modely, neboť se testuje pouze podobnost s původním modelem.

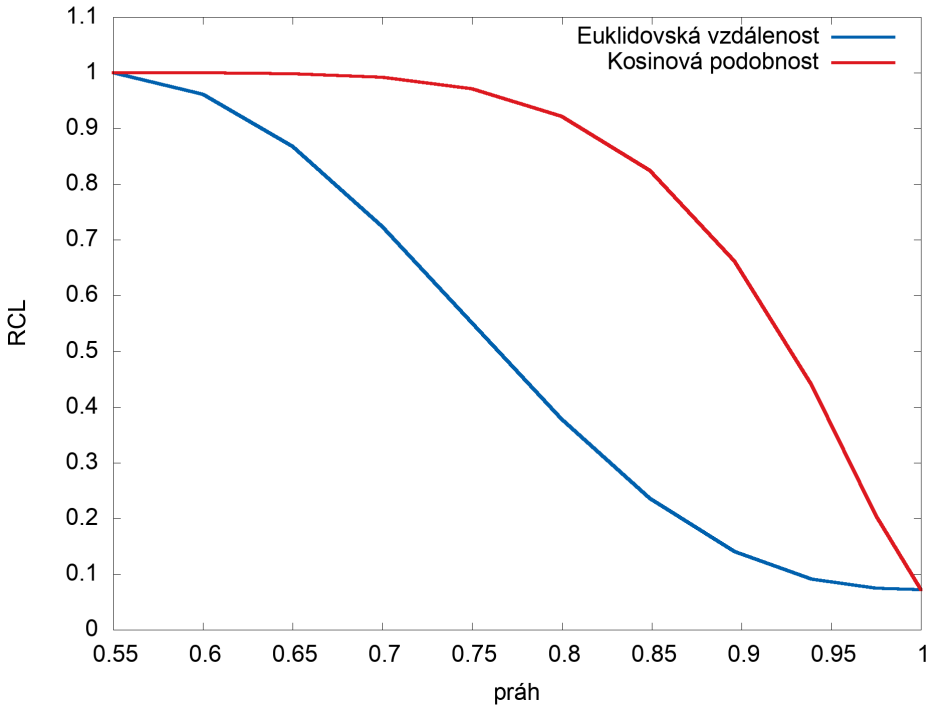
Všechny vybrané modely jsou nejprve náhodně otočeny kolem osy x v intervalu {0;359}, poté kolem všech os. Následně se upravovala jejich velikost a to podobným způsobem. Nejprve se upravila šířka modelu podle osy x v intervalu {0.1;2}. Dále se vytvořil další model s upravenými šířkami všech tří os. Úprava souřadnic bodů — nebo-li translace — proběhla také nejprve jen podle osy x v intervalu {-2;2} a poté s náhodnými hodnotami ve stejném intervalu pro všechny osy. Nakonec se model náhodně upravil translací a rotací, rotací a škálováním, translací a škálování a nakonec všemi třemi úpravami. Pro lepší porovnání a přesnější zjištění vlivu deformace na deskriptory se budou testovat nedeformované modely zvětšené o dvojnásobek a s náhodným nastavením škály v ose z různých intervalů. Osa x je škálovaná v intervalu {0.1;1}, osa x {1.1;2}, osa z {2;3}. Příklad úpravy jednoho vybraného modelu zobrazuje obrázek 5.5.

5.2 Srovnání porovnávacích metrik

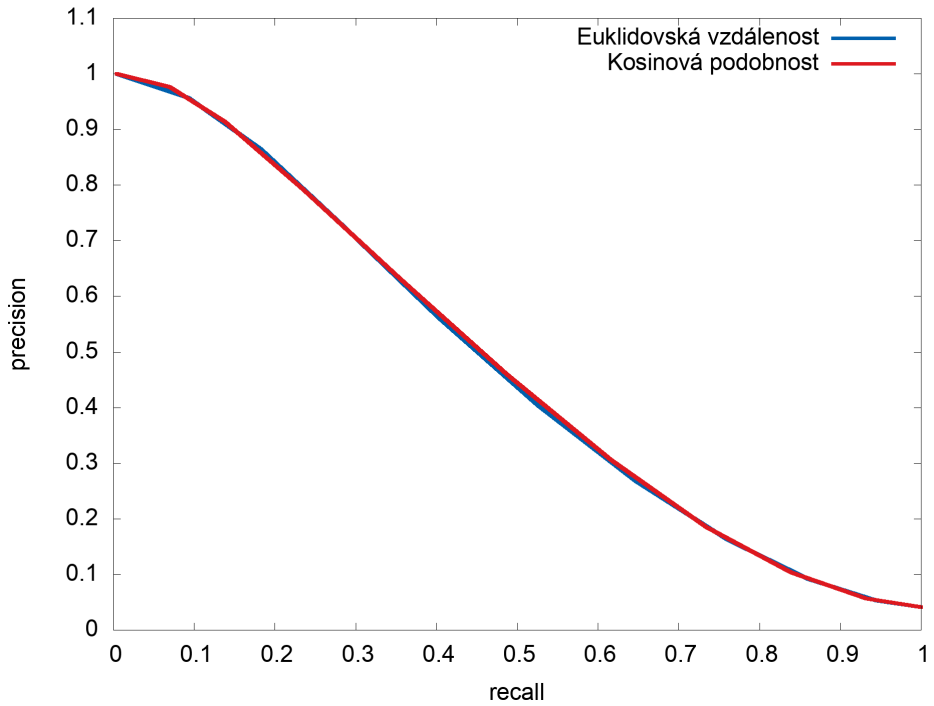
Porovnávací metriky se používají na vyhodnocení podobnosti dvou deskriptorů získaných z dvou modelů. Hlavní rozdíl mezi *Kosinovou podobností* a *Euklidovskou vzdáleností* je rozptyl hodnot vypočítaných podobností. Ten má největší vliv, pokud je třeba vyhledávat modely za základě prahové hodnoty podobnosti modelu. Tento rozptyl znázorňuje graf 5.6 a je z něj patrné, že *Kosinová podobnost* vykazuje jiný průběh křivky recall v závislosti na prahu. Porovnáváním Kosinové podobnosti dosáhne funkce *RCL* hodnoty 1 při prahu cca 0.74. Porovnání metodou *Euklidovské vzdálenosti* při prahu cca 0.55. Z toho vyplývá, že Euklidovská vzdálenost vrátí všechny správný výsledky při mnohem nižším prahu, než Kosinová podobnost. Výběr metriky porovnání má tedy zásadní vliv na výsledky porovnávání na základě prahové hodnoty.



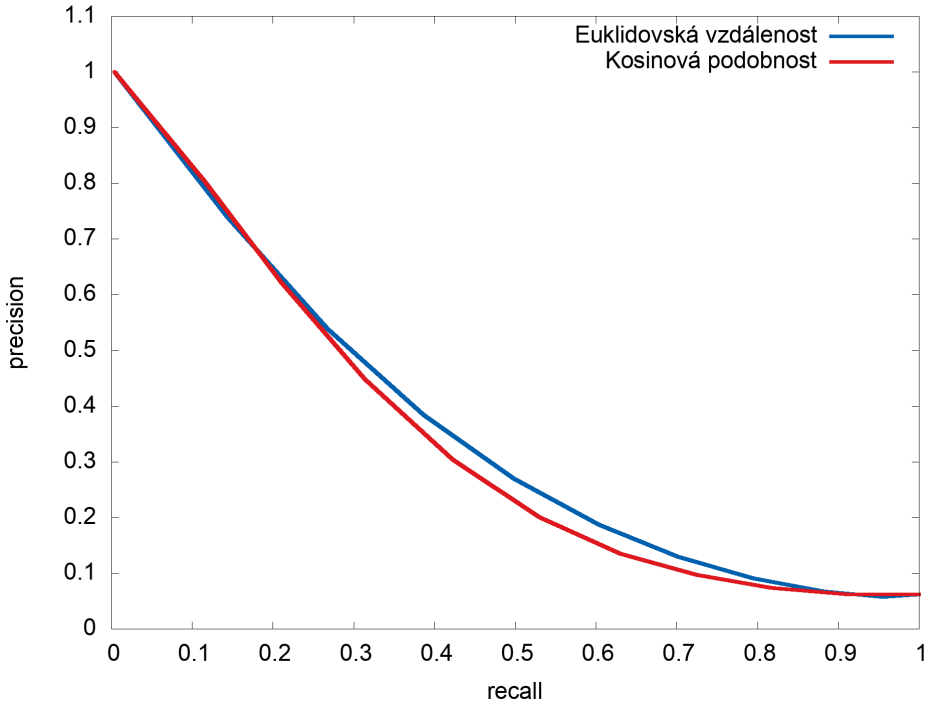
Obrázek 5.5: Příklad všech modifikací jednoho vybraného modelu.



Obrázek 5.6: Graf znázorňující průběh funkce *recall* v závislosti na prahu podobnosti. Kosinová podobnost dosahuje hodnoty 1 mnohem dříve, než Euklidovská vzdálenost. Práh byl postupně snižován a z výsledné množiny P vypočítána funkce recall.



Obrázek 5.7: Rozdíl v precision-recall křivce metody RISH při použití dvou porovnávacích metrik. Průběh křivek je téměř souměrný.



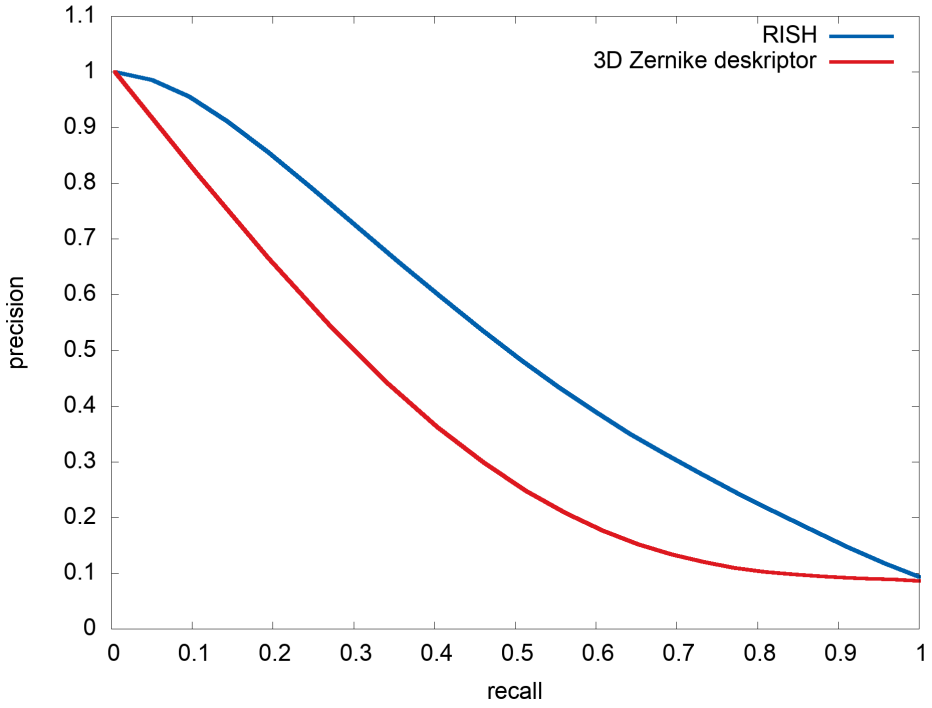
Obrázek 5.8: Rozdíl v precision-recall křivce 3D Zernike descriptors při použití dvou porovnávacích metrik. Pro Euklidovskou vzdálenost jsou výsledky nepatrně lepší.

V případě deskriptoru Rotation Invariant Spherical Harmonics (RISH) tento rozptyl nemá vliv na vyhledávání, kdy výsledkem je nejpodobnějších N modelů. Jak ukazuje graf 5.7, průběh precision-recall křivky je pro obě metriky stejný, stejně jako průběh funkce MAP. Výběr porovnávací metriky u tohoto deskriptoru tedy nemá vliv na kvalitu vyhledávání.

U *3D Zernike descriptors* je výběr porovnávací metriky pro nejpodobnějších N modelu důležitý. Z grafu 5.8 je zřejmé, že při použití Euklidovské vzdálenosti pracuje deskriptor o něco přesněji než v případě použití Kosinovy podobnosti.

Porovnání pro oba deskriptory probíhalo a na datových sadách Database_35 a Furnitude_11. Z datové sady Furnitude_11 se všechny modely použily jako dotaz na vyhledávání v databázi Database_35. Pro každou změnu prahové hodnoty byla pro každý dotaz vypočítána funkce precision a recall, které se poté pro jednotlivé prahy zprůměrovaly provedeným počtem dotazů. Výsledkem je precision-recall graf, kde na ose x jsou hodnoty *recall* a na ose y hodnoty *precision*.

Shrnutím poznatků se tedy dá říct, že *Kosinová podobnost* má menší rozptyl vypočítaných podobností než *Euklidovská vzdálenost*, tento rozdíl ale v případě deskriptoru RISH nemá vliv na přesnost deskriptoru. V případě *3D Zernike deskciptor* jsou výsledky vyhledávání přesnější při použití Euklidovské vzdálenosti. Výpočet Kosinové podobnosti požaduje více operací, je tedy výpočetně náročnější než Euklidovská vzdálenost. Z výše uvedených důvodů bude pro další srovnání metod *RISH* a *3D Zernike descriptors* použita Euklidovská vzdálenost.



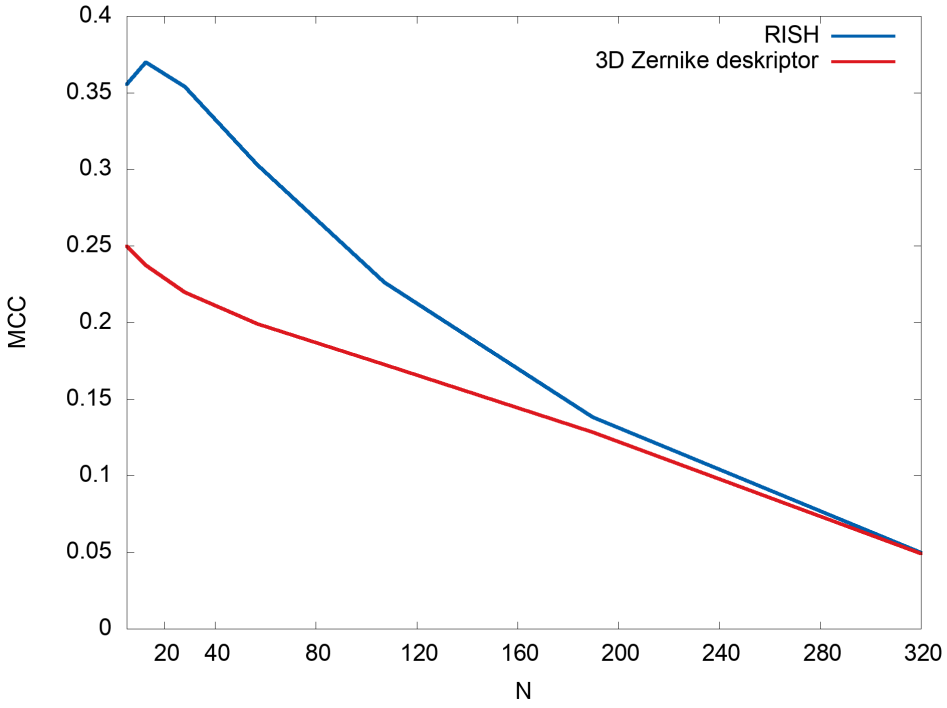
Obrázek 5.9: Srovnání přesnosti metod na extrakci deskriptorů. Metoda RISH vykazuje lepší výsledky precision po celém intervalu funkce recall.

5.3 Základní srovnání deskriptorů

Deskriptory *Rotation Invariant Spherical Harmonics (RISH)* a *3D Zernike descriptors* pracují na různém principu. Tento princip, který je popsaný v sekci 2.1, má výrazný vliv na přesnost vyhledávání, velikost vektoru i rychlost metod.

Na přesnost vyhledávání má i vliv nastavení rozlišení voxelizace modelu r a pro metodu *3D Zernike descriptors* i počet Zernikeho momentů M. Jelikož určení optimální hodnoty těchto proměnných není hlavní náplní této práce, protože se optimální hodnoty liší pro každou použitou datovou sadu, byla nejprve zvolena výchozí hodnota použitých knihoven. Ta byla následně upravena podle výsledků experimentů. Každý experiment změnil hodnoty proměnných r a N a následně porovnal určitý počet modelů. Ze získaných výsledků se vypočítala funkce Accuracy (ACC). Zvolená kombinace proměnných byla zvolena v závislosti na nejlepším výsledku funkce ACC. Pro deskriptor RISH byla v konečné fázi zvolena proměnná r rovna 64, pro *3D Zernike descriptors* proměnná r rovna 32 a proměnná M rovna 20. Tyto hodnoty se zdály velmi blízké optimálním hodnotám.

Přesnost vyhledávání se mění každou změnou počtu vrácených výsledků. Nejvhodnějším způsobem, jak srovnat přesnost dvou deskriptorů je *precision-recall* graf, kde je osa x *recall*(poměr všech správně vrácených výsledků ku počtu všech správných výsledků) a *precision*(poměr všech správně vrácených výsledků ku velikosti množiny výsledků označených systémem jako správné). Z tohoto grafu 5.9 je zřejmé, že metoda RISH pro libovolnou množinu vrácených výsledků vykazuje vždy vyšší přesnost vyhledávání. Nevýhoda této metody oproti *3D Zernike deskriptors* je vyšší paměťová náročnost z důvodu větší délky vypočítaného deskriptoru, a pomalejší extrahování deskriptoru z modelu. Větší délka vektoru má za následek i pomalejší porovnání dvou deskriptorů.



Obrázek 5.10: Srovnání funkce MCC dvou deskriptorů v závislosti na čísle N, kde N je počet nejpodobnějších prvků. Metoda RISH vykazuje mnohem lepší výsledky v první polovině intervalu.

Měření probíhalo na datových sadách Vehicles_11 a Database_35, kde všechny modely z datové sady Vehicles_11 se použili jako dotaz na vyhledávání v Database_35. Pro různé prahové hodnoty se vypočítaly funkce precision a recall, které se pro jednotlivé prahy zprůměrovaly počtem dotazů. Výsledkem je precision-recall graf.

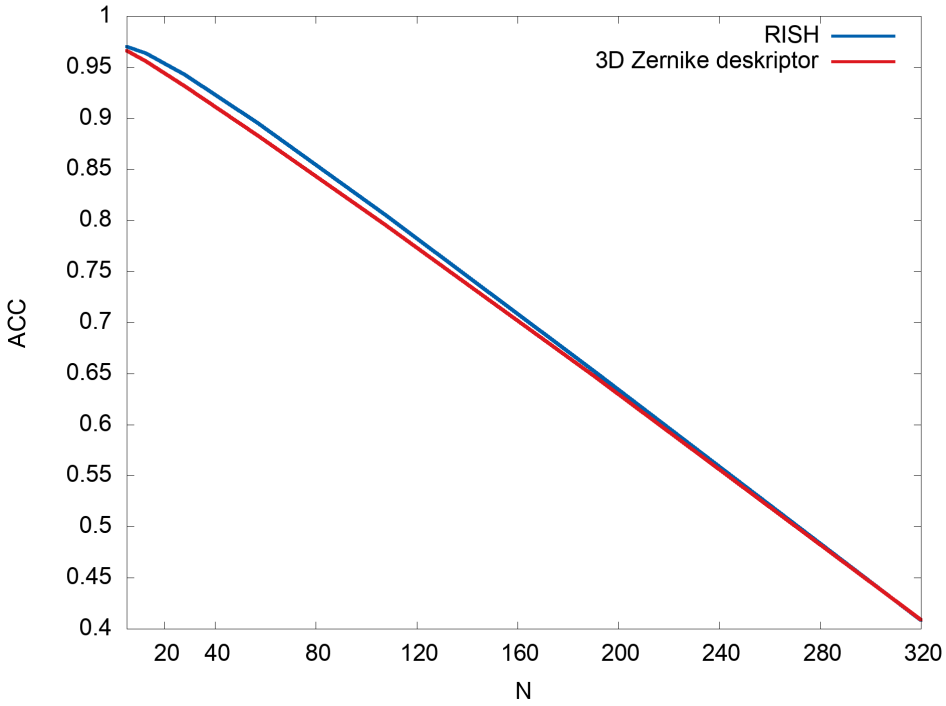
5.4 Vyhledání určitého počtu výsledků

Srovnávané metody budou nejspíše používané v systému, který bude hledat N nejpodobnějších výsledků. Na výsledky pro vrácení N podobných modelů se zaměřuje tato sekce.

5.4.1 Srovnání výsledků deskriptorů

Z grafů z předchozí sekce porovnávání porovnávacích metod je patrné, že metoda *3D Zernike descriptors* je obecně méně přesnější, než metoda RISH. Tento fakt můžeme srovnat i pomocí funkce MCC, jejíž průběh v závislosti na N vrácených modelech zobrazuje graf 5.10. Na grafu lze jasně vidět, že metoda RISH má koeficient větší, tudíž její výsledky v hledání jsou přesnější. Metoda *3D Zernike descriptors* má klesající tendenci, zatímco u metody RISH byla experimentálně nejvyšší hodnota naměřena při 21 vrácených modelech. Dá se tedy očekávat, že při vrácení prvních 21 modelů metodou RISH bude výsledek vykazovat nejlepší poměr správně vrácených výsledků ku chybně vrácených výsledků.

Výpočet MCC probíhal na základě průměru dotazů pro N vrácených modelů, kde dotazem byly postupně všechny modely databáze. Tento test se opakoval pro několik různých



Obrázek 5.11: Srovnání funkce ACC dvou metod na extrakci deskriptoru v závislosti na N vracených modelech. Metoda RISH vykazuje nepatrně lepší přesnost než metoda 3D Zernike descriptors.

hodnot prvních N výsledků.

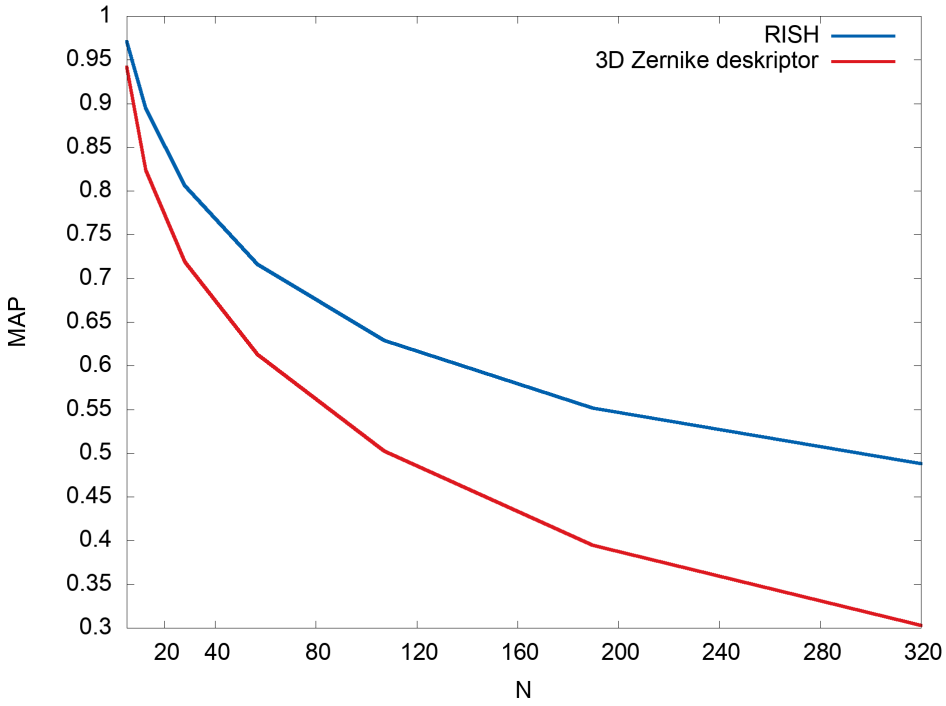
Zajímavé je i srovnání funkce Accuracy (ACC) obou metod. Z grafu 5.11 vyplývá, že i přes lepší výsledky metody RISH, vykazuje tato metoda jen o něco lepší funkci *accuracy*, tedy počet správně učených pozitivních výsledků a negativních výsledků. S přibývajícím počtem vrácených výsledků přesnost velmi rychle klesá. U metody RISH se u 20 výsledků opět nachází nepatrný schodek, který značí, že právě u tohoto čísla N vrácených výsledků můžeme dostávat nejlepší výsledky.

Výpočet ACC probíhal stejným způsobem a na stejné databázi jako MCC. Tedy vypočítala se funkce ACC pro všechny modely v databázi a hodnoty se poté zprůměrovaly. Všechny výpočty v téhle sekci probíhaly na datových sadách Furnitude_11 a Database_35, kde se modely z datové sady Furnitude_11 použili jako dotaz na vyhledávání v Database_35.

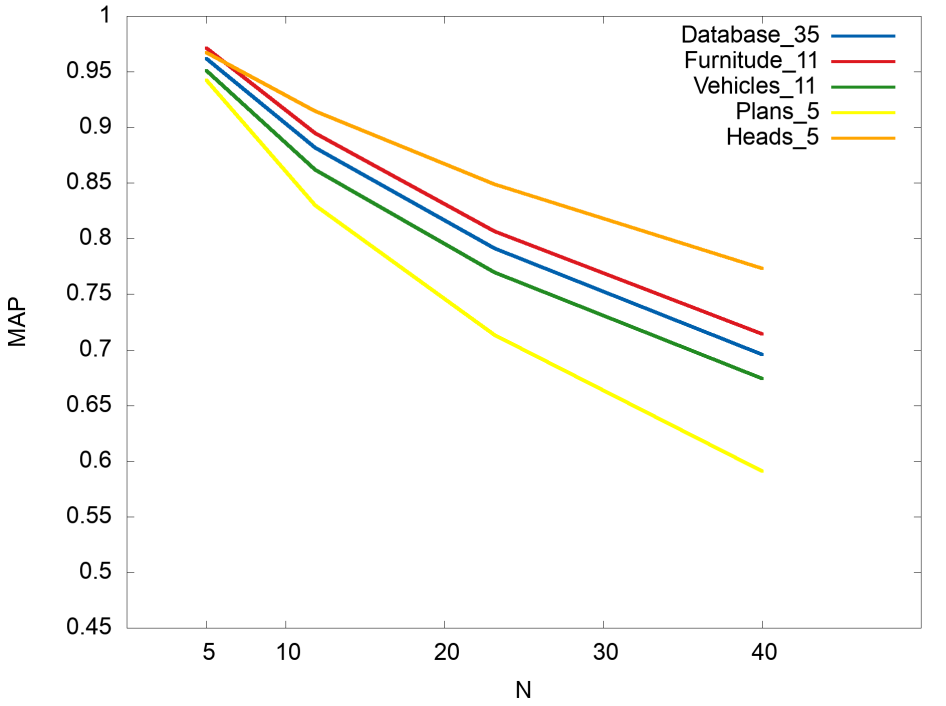
Velké rozdíly obou metod znázorňuje funkce MAP. Z grafu 5.12 zle vyčíst, že deskriptor RISH mnohem lépe řadí pravdivé výsledky. Pravdivé výsledky jsou vždy spíše na začátku množiny výsledků, zatímco u metody 3D Zernike descriptors jsou více rozptýleny.

5.4.2 Vliv podobnosti modelů na deskriptor RISH

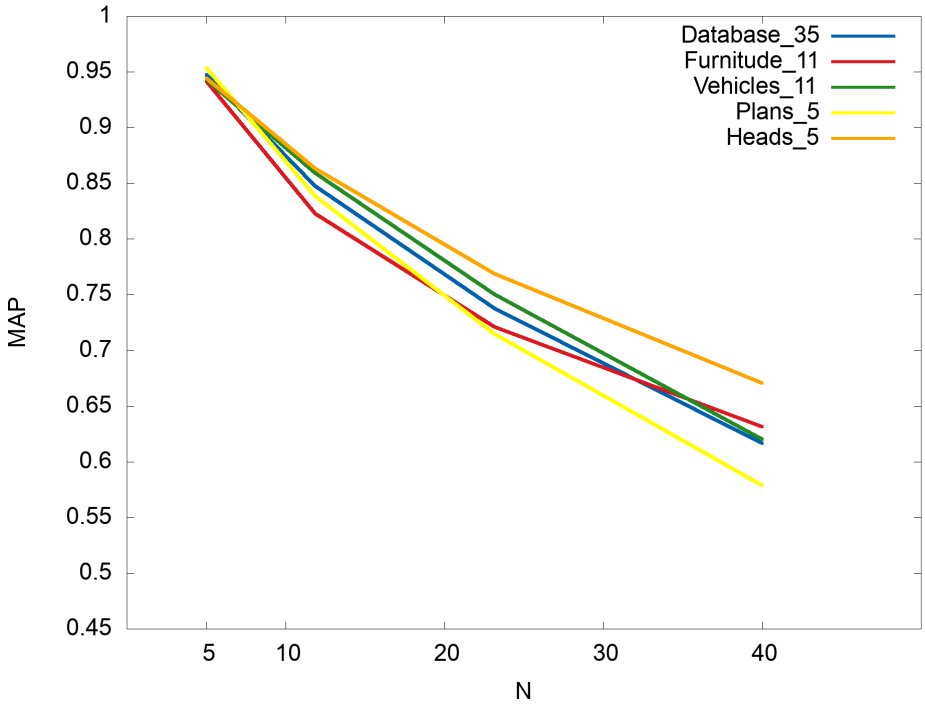
Různorodost vstupních dat má bezesporu vliv na výsledné metriky. Mezi N vracenými výsledky nás nejspíše bude zajímat počet vrácených správných výsledků a zda jsou spíše v popředí. Graf 5.13 názorně shrnuje chování výsledků metody na různě velkých databázích. Z grafu je patrné, že na metodu RISH nemá velký dopad typ databáze. Pro databáze Fur-



Obrázek 5.12: Srovnání funkce MAP dvou metod na extrakci deskriptoru v závislosti na N vracených modelech. Z grafu je patrné, že metoda RISH lépe řadí správné výsledky.



Obrázek 5.13: Rozdíly řazení správných výsledků metody RISH v množině výsledků v závislosti na podobnosti modelů v použitých datových sadách. Nejlepší výsledky vykazuje datová sada Heads_5.



Obrázek 5.14: Rozdíly řazení správných výsledků metody 3D Zernike descriptors v množině výsledků v závislosti na podobnosti modelů v použitých datových sadách. Vliv podobnosti modelů v sadách nemá na řazení takový vliv jako u metody RISH.

nitude_11, Vehicles_11 a Database_35, tedy pro velkou a střeně velkou databázi, je funkce MAP velmi podobná. Tedy pozice a počet pravdivě nalezených výsledků bude podobný. Datová sada Veicles_11, která byla klasifikována jako sada podobných modelů, má výsledky o něco horší.

Pro menší datové sady se ale výsledky liší více. Pro datovou sadu Heads_5, která byla charakterizovaná jako sada modelů s rozdílnými detaily, jsou výsledky lepší než pro tři dříve zmíněné sady. Pro datovou sadu Plants_5, která je typická velmi podobnými modely, jsou výsledky naopak horší.

5.4.3 Vliv podobnosti modelů na 3D Zernike descriptors

Jak ukazuje graf 5.14, i na metodu *3D Zernike descriptors* má vliv složení datových sad. Pro datové sady Furnitude_11, Vehicles_11 a Database_35 je průběh funkce MAP i zde velmi podobný. Oproti metodě RISH u této metody datová sada Vehicles_11 vykazuje mírně lepší výsledky než datová sada Furnitude_11. Rozdíl mezi hodnotami těchto tří sad a sady Plants_5 není tak velký jako v případě metody RISH. Z těchto poznatků lze usoudit, že metoda *3D Zernike descriptors* není tak náchylná ke složení datových sad, jako metoda RISH, přestože výsledky vyhledání nejsou tak přesné.

5.5 Experimenty s modely

V této části jsou vyhodnoceny výsledky pro upravené modely. Na základě výsledků bude patrné, jak moc deskriptory ovlivňuje, rotace, translace a škála. Experimenty byly provedeny

pro stejně upravených 100 modelů, aby se předešlo chybám. Uveden je průměr podobnosti všech upravených modelů k základnímu modelu.

Pro obě porovnávané metody platí, že porovnávaný model sám se sebou je si vždy podobný na 100%(vypočítaná podobnost 1). Jelikož pro tento experiment byla pro porovnávání použita Kosinová podobnost, dá se za úspěšné vyhledání změněného modelu považovat modely s průměrnou podobností nad 0.99.

5.5.1 Výsledky pro metodu RISH

Porovnávání vlastností probíhalo postupně na sadách, kde v každé sadě je obsažen původní model a modifikovaný model. Iterováním přes sadu se porovnává vždy jen modifikovaný model s původním a následně ze všech podobností se vypočítá aritmetický průměr.

Rotace

- **rotace podle osy x** - *0.99856*
- **rotace podle os xyz** - *0.99842*

Podobnost s původními modely je velmi vysoká, téměř 100%. Z toho můžeme usoudit, že metoda RISH je opravdu invariantní vůči rotacím, neboť všechny tři objekty byly vyhodnoceny jako velmi podobné.

Translace

- **translace po ose x** - *0.99889*
- **translace po osách xyz** - *0.99889*

I zde je s původními modely velmi velká podobnost. Tato metoda se dá označit tedy jako invariantní vůči translacím. Malá odchylka může být způsobena chybou reprezentace desetinného čísla ve vektoru.

Škála

- **škála podle osy x** - *0.95311*
- **škála podle os xyz** - *0.90713*
- **škálování na dvojnásobnou velikost** - *0.99889*
- **rozdílné intervaly** - *0.89336*

Je zřejmé, že škálováním modelu i deformací dostaneme rozdílně vypadající model. V případě škály ani vyšší počet detailů nepomohl k přesnější shodě modelů. Škálování modelu má tedy vliv na porovnávání touto metodou. Pro lepší srovnání jsem se rozhodl udělat ještě další dva testy škály. Jeden test se zabývá zvětšenými modely přesně o dvojnásobek. Druhý test nastavuje rozdílné intervaly pro všechny osy.

Podobnost dvojnásobně zvětšených modelů je téměř 100%. Resp. se jedná o stejnou průměrnou podobnost jako v případě translace. Souměrné zvětšení modelu na deskriptor RISH tedy nemá vliv.

V případě rozdílných intervalů se průměrná podobnost opět snížila. Upravením intervalů ze zvýšila i míra deformace modelů a to mělo za následek dalšího snížení podobnosti. Tento efekt je žádoucí, neboť splňuje základní požadavek na deskriptory modelů — deskriptory dvou různých modelů se liší více než deskriptory podobných modelů.

Kombinace modifikací

- **rotace + translace** - *0.99840*
- **rotace + škála** - *0.93283*
- **translace + škála** - *0.92758*
- **rotace + translace + škála** - *0.93229*

Výsledky kombinace modifikací modelů ukazují, že i kombinace rotace a translace nemá na výsledné deskriptory vliv. I zde se dle očekávání při deformování modelů pomocí škály snižuje podobnost modelů ve všech kombinacích.

5.5.2 Výsledky pro metodu 3D Zernike descriptors

Tato sekce se zabývá vyhodnocením experimentů pro metodu *3D Zernike descriptors* a následným srovnáním s metodou RISH. Z důvodu přesnějšího srovnání metod byly experimenty provedeny na stejné datové sadě a stejným způsobem.

Rotace

- **rotace podle osy x** - *0.94567*
- **rotace podle os xyz** - *0.94794*

Průměrná podobnost deskriptorů této metody oproti původním modelům se velmi liší. Tento typ deskriptoru tedy není vhodný k porovnávání modelů se změněnou rotací, dá se označit jako za neinvariantní vůči rotacím. Podle dosažených výsledků se zdá, že rotace podle os *xyz* nemá na na deskriptor větší vliv, než rotace podle jedné osy.

Translace

- **translace po ose x** - *0.99701*
- **translace po osách xyz** - *0.99701*

Zde je podobnost s původními modely velmi vysoká. V obou případech se jedná o stejnou průměrnou podobnost, dá se tedy říct, že translace na tento deskriptor nemá vliv, neboť modely si v obou případech byli podobné téměř na 100%. Odchylka, která je větší v porovnání s výsledky použití deskritoru RISH, může být způsobena rozdílnou délkou vektoru.

Škála

- **škála podle osy x** - *0.94378*
- **škála podle os xyz** - *0.90097*

- **škálování na dvojnásobnou velikost** - *0.99701*
- **změna intervalů** - *0.84938*

Vliv stejné deformace má na tento deskriptor o něco větší vliv, než na deskriptor RISH. V obou případech je výsledná podobnost menší než v případě deskriptoru RISH. I pro tento deskriptor je zajímavý vliv souměrného zvětšení modelu a různá úprava intervalů pro osy *xyz*. I zde byla použita stejně modifikovaná sada jako v případě deskriptoru RISH, tedy všechny modely jsou zvětšeny na dvojnásobek velikosti a interval škály pro osu x byl zvolen z intervalu $\{0.1;1\}$, pro osu y z intervalu $\{1.1;2\}$ a pro osu z z intervalu $\{2;3\}$.

Podobnost zvětšeného modelu je stejná jako v případě porovnávání translace modelu. Lze konstatovat, že i u tohoto deskriptoru nemá velikost modelu na porovnávání vliv.

Podobnost stejně deformovaných modelů je ale u tohoto deskriptoru mnohem nižší, než u deskriptoru RISH. Tento deskriptor i přes zachovalé detaily modelu vyhodnocuje rozdílné modely s menší podobností.

Kombinace modifikací

- **rotace + translace** - *0.95147*
- **rotace + škála** - *0.90911*
- **translace + škála** - *0.92303*
- **rotace + translace + škála** - *0.91574*

Kombinace modifikací podle očekávání vykazuje nižší pravděpodobnost než metoda RISH. Hlavním důvodem je neinvariantnost tohoto deskriptoru.

Kapitola 6

Závěr

Výsledkem této bakalářské práce je návrh a implementace jednoduché sady nástrojů, která poskytuje jednoduchý prototyp vyhledávacího systému nad databázemi modelů dvou vybraných deskriptorů *Rotation Invariant Shperical Harmonics* a *3D Zernike deskritor*. Tato sada nástrojů kromě vyhledávání poskytuje i funkce na zhodnocení kvality vyhledávání. Na základě těchto funkcí bylo provedeno srovnání obou deskriptorů.

Porovnávané deskriptory jsou rozšířením deskriptoru Spherical Harmonics, který patří mezi nejpoužívanější. Lepších výsledků dosahoval *Rotation Invariant Spherical Harmonics* deskriptor, který je zároveň invariantní vůči rotacím, translacím a rozdílné velikosti modelu, avšak větší velikost vektoru deskriptoru měla vliv na snížení rychlosti porovnávání. *3D Zernike descriptors* využívající 3D Zernikeho polynomy byl rychlejší, méně paměťově náročný, ale méně přesný. Zároveň není invariantní vůči rotacím, pouze vůči translacím a změně velikosti modelu.

Dále byli porovnány dvě porovnávací metriky, *Kosinová podobnost* a podobnost založená na *Euklidovské vzdálenosti*. Během experimentů bylo zjištěno, že výběr porovnávací metriky nemá vliv na přesnost deskriptoru *Rotation Invariant Shperical Harmonics* žádný vliv. *3D Zernike descriptors* pracuje lépe s *Euklidovskou vzdáleností*. Kosinová podobnost je výpočetně náročnější než Euklidovská vzdálenost.

Porovnání těchto metod může být v budoucnu využito při navrhování vyhledávacího systému, kdy tato práce přinese podrobnější srovnání chování dvou metod na porovnávání deskriptorů a dvou porovnávacích metrik. Sada nástrojů může být dále rozšířena o další deskriptory modelů, nebo o další metriky kvality vyhledávání a poté použita jako vyhodnocovací nástroj pro kvalitu vyhledávání na různých databázových sadách modelů.

Literatura

[1] Funkhouser, T.; Min, P.; Kazhdan, M.; aj.: A Search Engine for 3D Models. Princeton University, 2002.
URL https://www.cs.princeton.edu/~funk/tog03.pdf

[2] Kazhdan, M.; Funkhouser, T.; Rusinkiewicz, S.: *Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors*. Princeton University, 2003.
URL http://www.cs.jhu.edu/~misha/MyPapers/SGP03.pdf

[3] Kotásek, L.: Systém pro doporučení vhodné podobnostní míry. Masarykova univerzita Fakulta informatiky, 2014, diplomová práce.
URL https://is.muni.cz/th/256433/fi_m/System_pro_doporuceni_vhodne_podobnostni_miry.pdf

[4] Novotni, M.; Klein, R.: Shape retrieval using 3D Zernike descriptors. Computer Aided Design, 2004.
URL http://cg.cs.uni-bonn.de/en/publications/paper-details/novotni-2004-shape/

[5] W. H., T., Johan nad Remco; Veltkamp, C.: *A Survey of Content Based 3D Shape Retrieval Methods*. nstitute of Information and Computing Sciences, Utrecht University, 2007.
URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.8133&rep=rep1&type=pdf

[6] Zhang, L.; João da Fonseca, M.; Ferreira, A.: *Survey on 3D Shape Descriptors*. Instituto Superior Técnico, Universidade de Lisboa, 2004.
URL http://web.ist.utl.pt/alfredo.ferreira/publications/DecorAR-Surveyon3DShapedescriptors.pdf

Příloha A

Návod na použití skriptu

Kapitola 4.3 popisuje skript po implementační a funkční stránce. Tato příloha slouží jako dokumentace hlavního skriptu a návod na vyhledávání objektů v libovolné databázi. Skript je psaný v Pythonu 3.6.3 na platformě Windows, kompilované knihovny a nástroje jsou určeny taktéž pro platformu Windows 64-bitový systém. Celý projekt byl vyvíjen ve Visual Studiu 2017 od společnosti Microsoft.

A.0.1 Argumenty skriptu

Hlavní skript nese název `SreachScript.py`, lze jej spustit v příkazovém řádku. Avšak ke správnému vykonávání funkce potřebuje přítomnost kompilovaných knihoven a nástrojů v adresáři. Výjimku tvoří nástroj *Blender*, ke kterému se nastavuje cesta ve skriptu.

```
   ./SearchScript.py [--makePLY][--makeImgs][--makeBINVOX][--makeSHD]
[--makeZDD][--prepareData][--cmpSH FILENAME][--cmpZD FILENAME][--calcPRG PATHDATA
FILENAME METHOD][--calcTProp PATHDATA FILENAME METHOD][--calcProp PATHDATA FILENAME
METHOD]
[--calcAVS METHOD][--modData ACTION]
```

Parametr	**Popis parametru**
`makePLY`	Iteruje nad celou databází modelů a vytvoří novou databázi modelů ve formátu PLY.
`makeImgs`	Pro každý model databáze vygeneruje náhled modelu.
`makeBINVOX`	Modely nové databáze rasterizuje do binární mřížky.
`makeSHD`	Vytvoří databázi *Rotation Invariant Spherical harmonics* deskriptorů.
`makeZDD`	Vytvoří databázi *3D Zernimeho deskriptorů*.
`prepareData`	Spustí všechny předešlé akce.
`cmpSH FILENAME`	Porovná soubor FILENAME se RISH databází a zobrazí n výsledků.
`cmpZD FILENAME`	Porovná soubor FILENAME se ZD databází a zobrazí n výsledků.
`calcPRG PATHDATA FILENAME METHOD`	Vypočítá body precision-recall grafu a uloží je do FILENAME. PATHDATA určuje cestu k sadové sadě, která se bude vyhledávat.
`calcTProp PATHDATA FILENAME METHOD`	Vypočítá metriky pro celou databázi pro prahy uložené v TRESHOLDS. Výsledek je uložen do souboru FILENAME. PATHDATA určuje cestu k sadové sadě, která se bude vyhledávat.
`calcNProp PATHDATA`	Vypočítá metriky pro celou databázi pro sadu N modelů

`FILENAME METHOD`	uloženou v LIST_N. Výstupem je soubor FILENAME. PATHDATA určuje cestu k sadové sadě, která se bude vyhledávat.
`calcAVS METHOD`	Vypočítá průměrnou podobnost pro modifikované modely v databázi ku jejich originálu. Výstupem je standardní výstup.
`modData ACTION`	Pro všechny modely z databáze vygeneruje jejich modifikovanou verzi v závislosti na ACTION

Parametry se dají psát v libovolném pořadí, skript je vždy vykoná v tom správném. METHOD může nabývat hodnot "SH" pro porovnání RISH deskriptorem nebo hodnotou "ZD" pro porovnání 3D Zernikeho deskriptorem.

Hodnoty ACTION

Hodnota	**Popis**
`rotateX`	Rotování podle osy X v intervalu {0;359}.
`rotateXYZ`	Rotování podle všech os v intervalu {0;359}.
`translateX`	Translace po ose X v intervalu {-2;2}.
`translateXYZ`	Translace po všech osách v intervalu {-2;2}.
`scaleX`	Škálování po ose X v intervalu {0.1;2}.
`scaleXYZ`	Škálování po všech osách v intervalu {0.1;2}.
`scaleXYZ2`	Zvětšení modelu na dvojnásobnou velikost.
`scaleXYZrand`	Škála po ose X v intervalu {0.1;1}, ose Y 1.1;2, ose Z 2;3.
`rotateTranslate`	Hodnoty rotateXYZ a translateXYZ.
`rotateScale`	Hodnoty rotateXYZ a scaleXYZ.
`scaleTranslate`	Hodnoty scaleXYZ a translateXYZ.
`rotateScaleTranslate`	Hodnoty rotateXYZ, scaleXYZ a translateXYZ.

A.0.2 Nastavení skriptu

Interní nastavení skriptu se provádí změnou hodnoty globální proměnné.

Název proměnné	**Popis vlivu hodnoty**
`ACCEPTABLE_FORMATS`	N-tice akceptovatelných formátů původní databáze a formátu porovnávaného modelu
`FIRST_N`	Nastavuje zobrazování n výsledků.
`FIRST_N_OFFSET`	Nastaví offset zobrazovaných výsledků.
`TRESHOLDS`	Seznam prahů pro výpočet precision-recall grafu.
`LIST_N`	Seznam N vrácených výsledků.
`COMPARE_METHOD`	Zvolení porovnávací metody. Hodnota "euclidCMP"volí Euklidovskou vzdálenost, jinak je zvolena Kosinova podobnost.
`PATH_DATABASE`	Nastavení cesty k původní databázi.
`PATH_PREDATABASE`	Nastavení cesty k nově připravené databázi.
`PATH_DATABASE_IMAGES`	Cesta do složky, kam se ukládají generované náhledy databáze.
`PATH_SHDFILES`	Cesta k databázi obsahující *Spherical harmonics* deskriptory.
`PATH_ZDFILES`	Cesta k databázi obsahující *3D Zernikeho* deskriptory.
`PATH_RESULT_IMAGES`	Cesta do složky, kam se budou ukládat výsledky generování náhledů

	výsledku hledání.
`PATH_BLENDER`	Cesta ke spouštěcímu souboru nástroje Blender

Skript podporuje jak relativní, tak absolutní cesty. Dokáže se automaticky přizpůsobit jakékoliv změně bez nutnosti zasahovat do ostatního kódu.

Příloha B

Obsah DVD

Součásti práce je DVD s finální sadou nástrojů s použitými datovými sadami a modifikovanými modely.

B.0.1 Struktura složek

Soubor/složka	Popis
`Classes`	Složka obsahující datové sady.
`- Database_35`	
`- Furnitude_11`	
`- Heads_5`	
`- Plans_5`	
`- Vehicles_11`	
`Documents`	Složka s technickou zprávou a plakátem.
`Modified_models`	Složka obsahující modifikované modely.
`- rotateScale`	
`- rotateScaleTranslate`	
`- rotateTranslate`	
`- rotateX`	
`- rotateXYZ`	
`- scaleTranslate`	
`- scaleX`	
`- scaleXYZ`	
`- scaleXYZ2`	
`- scaleXYZrand`	
`- translateX`	
`- translateXYZ`	
`Result`	Složka pro generování obrázků výsledku vyhledávání v databázi.
`binvox.exe`	Nástroj pro rasterizování modelu do 3D mřížky.
`BlenderRender.py`	Skript předávaný nástroji Blender pro vygenerování náhledů.
`libfftw3-3.dll`	Knihovna pro ShapeDescriptor.exe.
`libfftw3f-3.dll`	Knihovna pro ShapeDescriptor.exe
`meshconv.exe`	Nástroj pro konverzi modelů.
`ModificModels.py`	Skript předávaný nástroji Blender pro modifikování modelů.
`SearchScript.py`	Hlavní skript.
`ShapeDescriptor.exe`	Kompilovaná knihovna pro extrakci RISH deskriptoru.

`ZernikeDescriptor.exe` Kompilovaná knihovna pro extrakci 3D Zernike Deskriptor.

B.0.2 Použití

Všechny akce se provádí přes hlavní skript `SearchScriptpy`. Všechny datové sady jsou již předpřipravené na použití. Výchozí nastavení vede na databázi Database_35. Změnu databáze docílíte změnou globální proměnné `PATH_PREDATABASE` na jinou složku. Skript začne automaticky vyhledávat v nově zvolené databázi. Při změně na modifikované modely musíte měnit hodnotu globální proměnné `PATH_DATABASE` na hodnotu "Modified_models". Do `PATH_PREDATABASE` poté nastavíte název složky konkrétní databáze.

Měření metrik databáze se ukládá do vámi zvoleného souboru. Při měření metrik se na standardní výstup postupně vypisují zpracovávané modely.

Příloha C

Tabulky výsledků metody RISH

Při použití Euklidovské vzdálenosti

Sada	Funkce	Práh podobnosti					
		0.98	0.96	0.93	0.8	0.75	0.7
Furnitude_11	ERR	0.03132	0.03131	0.0312	0.03189	0.08457	0.27838
	ACC	0.96868	0.96869	0.9688	0.96811	0.91543	0.72162
	FPR	0.0	0.0	0.0	0.00968	0.07334	0.28064
	MAP	1.0	1.0	1.0	0.92238	0.74123	0.56769
	PRC	1.0	1.0	1.0	0.71011	0.32405	0.1149
	RCL	0.07354	0.07384	0.07605	0.31073	0.56949	0.78103
	MCC	0.25941	0.26004	0.26374	0.41536	0.33369	0.14852
Vehicles_11	ERR	0.03558	0.03552	0.03541	0.0569	0.17057	0.40276
	ACC	0.96442	0.96448	0.96459	0.9431	0.82943	0.59724
	FPR	0.0	0.0	0.0	0.0439	0.17306	0.4172
	MAP	1.0	1.0	1.0	0.73192	0.57953	0.53828
	PRC	1.0	1.0	1.0	0.38491	0.17346	0.08692
	RCL	0.08436	0.08566	0.08992	0.58725	0.87707	0.96402
	MCC	0.27043	0.27295	0.27852	0.39982	0.19888	0.04794
Heads_5	ERR	0.04451	0.04451	0.04442	0.03369	0.0507	0.18373
	ACC	0.95549	0.95549	0.95558	0.96631	0.9493	0.81627
	FPR	5e-05	5e-05	5e-05	0.00375	0.03586	0.18761
	MAP	0.99432	0.99432	0.99432	0.95651	0.81802	0.67684
	PRC	0.98864	0.98864	0.98864	0.78405	0.4773	0.21928
	RCL	0.06098	0.06098	0.0625	0.34491	0.61416	0.86907
	MCC	0.22545	0.22545	0.2288	0.48228	0.49146	0.31843
Plants_5	ERR	0.04382	0.04382	0.04382	0.04539	0.07618	0.21256
	ACC	0.95618	0.95618	0.95618	0.95461	0.92382	0.78744
	FPR	0.0	0.0	0.0	0.004	0.04303	0.19905
	MAP	1.0	1.0	1.0	0.94102	0.75073	0.51092
	PRC	1.0	1.0	1.0	0.74977	0.39362	0.20185
	RCL	0.05463	0.05463	0.05463	0.09941	0.23215	0.48514
	MCC	0.22134	0.22134	0.22134	0.2367	0.22941	0.2048

Sada	Funkce	TOP N výsledků				
		5	10	20	40	80
Database_35	ERR	0.03686	0.03892	0.04647	0.07186	0.13811
	ACC	0.96314	0.96108	0.95352	0.92813	0.86189
	MAP	0.96180	0.88870	0.79232	0.69566	0.61900
	RCL	0.20061	0.29273	0.42430	0.57331	0.70463
	FRP	0.00328	0.00931	0.02351	0.05926	0.14978
	MCC	0.33712	0.36056	0.38204	0.36480	0.30173
Furnitude_11	ERR	0.02992	0.03213	0.04164	0.07032	0.14047
	ACC	0.97008	0.96787	0.95836	0.92968	0.85953
	FPR	0.00324	0.00937	0.02427	0.05912	0.1355
	MAP	0.97106	0.90389	0.80515	0.71417	0.62038
	PRC	0.67385	0.53	0.39231	0.26115	0.15433
	RCL	0.20827	0.30891	0.4411	0.57931	0.69077
	MCC	0.35555	0.38124	0.38699	0.33868	0.21864
Vehicles_11	ERR	0.03414	0.03626	0.04451	0.06869	0.13169
	ACC	0.96586	0.96374	0.95549	0.93131	0.86831
	FPR	0.00322	0.00929	0.02353	0.05607	0.12897
	MAP	0.95076	0.87344	0.77074	0.67411	0.59349
	PRC	0.67692	0.53385	0.41038	0.29923	0.19644
	RCL	0.24421	0.34738	0.47946	0.64216	0.81377
	MCC	0.37202	0.36396	0.365	0.35465	0.28397
Heads_5	ERR	0.04129	0.03949	0.03852	0.05807	0.12352
	ACC	0.95871	0.96051	0.96148	0.94193	0.87648
	FPR	0.00233	0.00641	0.01597	0.04647	0.12143
	MAP	0.96716	0.9228	0.85201	0.77327	0.70667
	PRC	0.76818	0.68068	0.60284	0.42528	0.25156
	RCL	0.1922	0.30358	0.50007	0.69587	0.82507
	MCC	0.3567	0.42451	0.51616	0.49307	0.37917
Plants_5	ERR	0.04585	0.05065	0.06224	0.0899	0.15354
	ACC	0.95415	0.94935	0.93776	0.9101	0.84646
	FPR	0.00508	0.01265	0.02884	0.06361	0.13752
	MAP	0.94223	0.83146	0.71714	0.59094	0.49615
	PRC	0.49773	0.375	0.28807	0.21562	0.15256
	RCL	0.11951	0.17019	0.24828	0.36523	0.50368
	MCC	0.22205	0.22386	0.22961	0.23012	0.21032

Při použití Kosinové podobnosti

Sada	Funkce	Práh podobnosti					
		0.98	0.96	0.93	0.8	0.75	0.7
Furnitude_11	ERR	0.0284	0.04439	0.14641	0.81328	0.90531	0.9469
	ACC	0.9716	0.95561	0.85359	0.18672	0.09469	0.0531
	FPR	0.00132	0.02616	0.14077	0.84044	0.9362	0.97942
	MAP	0.98402	0.86339	0.65801	0.47112	0.4645	0.46368
	PRC	0.89662	0.50489	0.21218	0.03964	0.03579	0.03408
	RCL	0.17948	0.40402	0.66943	0.97761	0.99695	1.0
	MCC	0.36659	0.38349	0.26132	0.01338	0.00885	0.00816
Vehicles_11	ERR	0.03248	0.09411	0.25645	0.84442	0.91816	0.94952
	ACC	0.96752	0.90589	0.74355	0.15558	0.08184	0.05048
	FPR	0.0088	0.08784	0.26434	0.87718	0.95408	0.9867
	MAP	0.9095	0.65583	0.55681	0.52626	0.52626	0.52626
	PRC	0.65551	0.26808	0.12407	0.04318	0.03931	0.03797
	RCL	0.34936	0.72234	0.92912	1.0	1.0	1.0
	MCC	0.43367	0.33696	0.12023	0.00952	0.00865	0.00835
Heads_5	ERR	0.03898	0.03352	0.08959	0.69827	0.8358	0.90799
	ACC	0.96102	0.96648	0.91041	0.30173	0.1642	0.09201
	FPR	0.00108	0.00954	0.08376	0.73303	0.87691	0.95236
	MAP	0.98894	0.91407	0.73281	0.62078	0.62078	0.62078
	PRC	0.91128	0.68287	0.34362	0.06657	0.05348	0.04882
	RCL	0.19364	0.45513	0.76855	1.0	1.0	1.0
	MCC	0.37743	0.52083	0.45336	0.01389	0.011	0.00999
Plants_5	ERR	0.04371	0.05105	0.11831	0.72459	0.86232	0.92548
	ACC	0.95629	0.94895	0.88169	0.27541	0.13768	0.07452
	FPR	0.00044	0.01167	0.09277	0.75584	0.90215	0.96929
	MAP	0.99905	0.8939	0.62368	0.30653	0.29526	0.29188
	PRC	0.93333	0.59096	0.31322	0.0615	0.0502	0.04684
	RCL	0.06423	0.13451	0.34133	0.92898	0.97394	0.99021
	MCC	0.2277	0.23195	0.23541	0.05887	0.02252	0.01289

Příloha D

Tabulky výsledků metody 3D Zernike Deskriptor

Při použití Euklidovské vzdálenosti

Sada	**Funkce**	**Práh podobnosti**					
		0.98	0.96	0.93	0.8	0.75	0.7
Furnitude_11	ERR	0.03317	0.05072	0.10438	0.56473	0.70733	0.81732
	ACC	0.96683	0.94928	0.89562	0.43527	0.29267	0.18268
	FPR	0.00355	0.02583	0.08874	0.58258	0.73139	0.84518
	MAP	0.94894	0.82463	0.65829	0.32955	0.29151	0.28583
	PRC	0.81178	0.60745	0.33518	0.05229	0.0446	0.03967
	RCL	0.11083	0.20648	0.39097	0.89873	0.97621	0.99719
	MCC	0.26428	0.2549	0.24423	0.05503	0.01775	0.01079
Vehicles_11	ERR	0.03646	0.05487	0.12089	0.61822	0.73493	0.829
	ACC	0.96354	0.94513	0.87911	0.38178	0.26507	0.171
	FPR	0.00271	0.02801	0.10508	0.63996	0.76321	0.86162
	MAP	0.97079	0.80399	0.59982	0.3537	0.34091	0.33965
	PRC	0.82667	0.44792	0.21832	0.05675	0.04917	0.04279
	RCL	0.15013	0.30738	0.49431	0.93933	0.97682	0.98769
	MCC	0.29115	0.25928	0.22632	0.04984	0.02168	0.00751
Heads_5	ERR	0.04451	0.04407	0.04601	0.29197	0.43343	0.59954
	ACC	0.95549	0.95593	0.95399	0.70803	0.56657	0.40046
	FPR	5e-05	9e-05	0.00638	0.29746	0.45118	0.62837
	MAP	0.99432	0.99432	0.93264	0.46071	0.42913	0.418
	PRC	0.98864	0.97727	0.76695	0.16369	0.12033	0.08541
	RCL	0.06044	0.07442	0.16901	0.83485	0.9392	0.98996
	MCC	0.22516	0.24463	0.3096	0.25557	0.14247	0.02404
Plants_5	ERR	0.04482	0.05105	0.0721	0.43182	0.57778	0.73313
	ACC	0.95518	0.94895	0.9279	0.56818	0.42222	0.26687
	FPR	0.00098	0.00764	0.03033	0.42774	0.58945	0.75812
	MAP	1.0	0.97078	0.90015	0.3053	0.25717	0.23439
	PRC	0.95159	0.84462	0.685	0.11446	0.09793	0.07954
	RCL	0.05312	0.06069	0.09469	0.55708	0.71943	0.83524
	MCC	0.20885	0.18733	0.16166	0.07671	0.06476	0.02021

Sada	Funkce	TOP N výsledků				
		5	10	20	40	80
Database_35	ERR	0.04038	0.04603	0.05899	0.08904	0.15385
	ACC	0.95961	0.95397	0.94101	0.91095	0.84614
	MAP	0.94741	0.85209	0.74338	0.61665	0.48748
	RCL	0.14135	0.18996	0.26307	0.36507	0.51897
	FRP	0.00504	0.01292	0.02300	0.06855	0.15907
	MCC	0.23656	0.22518	0.21982	0.20788	0.19682
Furnitude_11	ERR	0.03394	0.03973	0.0541	0.08696	0.15249
	ACC	0.96606	0.96027	0.9459	0.91304	0.84751
	FPR	0.00534	0.01334	0.03078	0.0678	0.14177
	MAP	0.94137	0.81199	0.71663	0.63148	0.49579
	PRC	0.46615	0.33385	0.23154	0.15385	0.11558
	RCL	0.15575	0.21319	0.28575	0.36367	0.53596
	MCC	0.2496	0.24124	0.22374	0.19238	0.192
Vehicles_11	ERR	0.03733	0.04281	0.05486	0.08372	0.15066
	ACC	0.96267	0.95719	0.94514	0.91628	0.84934
	FPR	0.00486	0.01271	0.02896	0.064	0.13898
	MAP	0.94304	0.87235	0.75972	0.61992	0.52077
	PRC	0.51231	0.36462	0.27692	0.20231	0.13529
	RCL	0.17172	0.2276	0.31461	0.4396	0.57028
	MCC	0.26567	0.25062	0.24366	0.22506	0.192272
Plants_5	ERR	0.04832	0.05536	0.06999	0.1028	0.16869
	ACC	0.95168	0.94464	0.93001	0.8972	0.83131
	FPR	0.00638	0.01513	0.03293	0.07041	0.14552
	MAP	0.95366	0.84229	0.72124	0.5785	0.43759
	PRC	0.37045	0.25341	0.18807	0.13239	0.10369
	RCL	0.08649	0.11222	0.15955	0.21872	0.33524
	MCC	0.15768	0.13971	0.13351	0.11528	0.11152
Heads_5	ERR	0.04358	0.04671	0.05517	0.07939	0.13942
	ACC	0.95642	0.95329	0.94483	0.92061	0.86058
	FPR	0.00354	0.01023	0.02477	0.05774	0.12982
	MAP	0.94414	0.87253	0.76811	0.67033	0.55184
	PRC	0.65	0.49432	0.38807	0.28778	0.20028
	RCL	0.16163	0.22774	0.33329	0.48032	0.66444
	MCC	0.29788	0.30382	0.32126	0.32323	0.2845

Při použití Kosinové podobnosti

Sada	Funkce	Práh podobnosti					
		0.98	0.96	0.93	0.8	0.75	0.7
Furnitude_11	ERR	0.0586	0.12531	0.2366	0.67442	0.80517	0.89259
	ACC	0.9414	0.87469	0.7634	0.32558	0.19483	0.10741
	FPR	0.03669	0.11328	0.23464	0.69573	0.83187	0.923
	MAP	0.79319	0.59513	0.46402	0.33278	0.3268	0.32114
	PRC	0.40868	0.18312	0.09977	0.04695	0.04	0.03615
	RCL	0.29472	0.49058	0.67118	0.94931	0.97598	0.99328
	MCC	0.27591	0.2304	0.16282	0.0323	0.01559	0.01003
Vehicles_11	ERR	0.08366	0.17679	0.29727	0.70376	0.81689	0.88971
	ACC	0.91634	0.82321	0.70273	0.29624	0.18311	0.11029
	FPR	0.06561	0.16884	0.30005	0.73063	0.84892	0.92487
	MAP	0.67669	0.53237	0.43269	0.34506	0.34297	0.34137
	PRC	0.23943	0.12032	0.08808	0.05049	0.04363	0.04025
	RCL	0.4395	0.60514	0.77061	0.97932	0.98958	0.99864
	MCC	0.24426	0.18847	0.14756	0.01952	0.01221	0.01033
Heads_5	ERR	0.05329	0.09928	0.19208	0.5902	0.74852	0.85593
	ACC	0.94671	0.90072	0.80792	0.4098	0.25148	0.14407
	FPR	0.02651	0.08632	0.19184	0.61865	0.7847	0.89745
	MAP	0.77207	0.62573	0.54001	0.4506	0.44976	0.44965
	PRC	0.50554	0.27662	0.17246	0.078	0.06024	0.0521
	RCL	0.40743	0.63826	0.78201	0.9929	0.99858	1.0
	MCC	0.38374	0.33727	0.28018	0.01992	0.01351	0.01065
Plants_5	ERR	0.05735	0.1056	0.20214	0.59514	0.72003	0.82345
	ACC	0.94265	0.8944	0.79786	0.40486	0.27997	0.17655
	FPR	0.0162	0.07243	0.18223	0.61272	0.74787	0.85881
	MAP	0.90089	0.59674	0.3821	0.21953	0.19934	0.19184
	PRC	0.47503	0.21082	0.12508	0.06868	0.05806	0.051
	RCL	0.08913	0.20141	0.39282	0.80123	0.8921	0.93478
	MCC	0.15447	0.12879	0.12665	0.07765	0.04875	0.02079